



UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO  
DEPARTAMENTO DE COMUNICAÇÕES

DISSERTAÇÃO DE MESTRADO

# ANÁLISE DE DESEMPENHO DA DISCIPLINA DE SERVIÇO “GENERALIZED PROCESSOR SHARING” SOB TRÁFEGO AUTO-SIMILAR

FLÁVIO DE MELO PEREIRA

## **Orientadores**

Prof. Dr. Dalton Soares Arantes  
*Faculdade de Engenharia Elétrica e de Computação, UNICAMP*

Prof. Dr. Nelson Luis Saldanha da Fonseca  
*Instituto de Computação, UNICAMP*

## **Banca examinadora**

Prof. Dr. Dalton Soares Arantes (Presidente, FEEC/UNICAMP)  
Prof. Dr. Otto Carlos Muniz Bandeira Duarte (COPPE-EE/UFRJ)  
Prof. Dr. Akebo Yamakami (FEEC/UNICAMP)  
Prof. Dr. Lee Luan Ling (FEEC/UNICAMP)

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

---

Campinas, 4 de março de 2002

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

P414a

Pereira, Flávio de Melo

Análise de desempenho da disciplina de serviço  
“generalized processor sharing” sob tráfego auto-similar  
/ Flávio de Melo Pereira.--Campinas, SP: [s.n.], 2002.

Orientadores: Dalton Soares Arantes e Nelson Luis  
Saldanha da Fonseca.

Dissertação (mestrado) - Universidade Estadual de  
Campinas, Faculdade de Engenharia Elétrica e de  
Computação.

1. Redes de computação. 2. Comutação de pacotes  
(Transmissão de dados). I. Arantes, Dalton Soares. II.  
Fonseca, Nelson Luis Saldanha. III. Universidade  
Estadual de Campinas. Faculdade de Engenharia Elétrica  
e de Computação. IV. Título.

# Resumo

Esta dissertação trata do problema da análise de desempenho da disciplina de serviço denominada *Generalized Processor Sharing* (GPS). Em particular, esta dissertação é baseada nos estudos de Parekh e Gallager, que demonstram que limitantes para o *backlog* e para o atraso em um servidor GPS podem ser obtidos quando o tráfego é regulado pelo algoritmo do Balde Furado. Este resultado foi, então, aplicado na análise de redes de servidores GPS com topologia arbitrária, sendo demonstrado que condições suficientes para garantir estabilidade podem ser estabelecidas caso a parametrização dos servidores pertença a uma classe denominada Tratamento de Sessões Relativamente Consistente (*Consistent Relative Session Treatment*, CRST). Além disso, foi demonstrado que, para uma rede GPS-CRST estável, limitantes para o *backlog* e para o atraso fim-a-fim podem ser obtidos utilizando um algoritmo baseado na caracterização do serviço oferecido pela rede às sessões por uma função denominada Curva de Serviço Universal.

Apesar de sua relevância, a análise proposta por Parekh e Gallager não leva em consideração a natureza auto-similar do tráfego em redes, cujos efeitos negativos no desempenho são reconhecidos. Nesta dissertação, a análise proposta por Parekh e Gallager para um servidor GPS isolado é estendida ao caso do tráfego auto-similar. Assume-se que o tráfego das sessões é regulado pelo algoritmo do Balde Furado Fractal, que é um mecanismo de policiamento adequado ao tratamento da auto-similaridade. É apresentado um algoritmo para o cálculo de limitantes de atraso e *backlog* para um servidor GPS isolado. Os resultados são, então, estendidos para uma rede de servidores GPS com topologia arbitrária, sendo demonstrado que as condições de estabilidade são, neste caso, equivalentes às aquelas obtidas por Parekh e Gallager para o caso do tráfego regulado pelo algoritmo do Balde Furado. Finalmente, o algoritmo baseado na Curva de Serviço Universal é estendido ao caso do tráfego auto-similar.

# Abstract

In this dissertation, the performance analysis of the Generalized Processor Sharing (GPS) service discipline is carried out. In particular, the analysis is based on the studies of Parekh and Gallager, who showed that, under certain conditions, bounds on individual session backlog and delay can be obtained if the traffic is Leaky Bucket regulated. Such a result was then applied in the analysis of GPS networks with arbitrary topology. The authors proved that stability conditions can be established if the traffic is Leaky Bucket regulated and the network servers operate under a broad class of assignments known as Consistent Relative Session Treatment (CRST). Moreover, an algorithm for the computation of bounds on end-to-end backlog and delay for a stable GPS-CRST network was introduced. This algorithm is based on a specific representation of the service offered to the sessions by the network, which is called Universal Service Curve.

Despite its importance, the analysis proposed by Parekh and Gallager does not take the self-similar nature of network traffic nor its negative impacts into consideration. In the present dissertation, the analysis proposed by Parekh and Gallager is extended to the self-similar traffic case. The traffic is assumed to be regulated by the Fractal Leaky Bucket algorithm, which is an adequate policing mechanism for self-similar traffic. An algorithm for the computation of bounds on the backlog and on the delay in a GPS server is introduced. Results are then extended to a network of GPS servers with arbitrary topology, and the stability of those networks is analyzed. Stability conditions are shown to be equivalent to those obtained by Parekh and Gallager for the Leaky Bucket regulated traffic. The algorithm based on the Universal Service Curve is also extended to the self-similar traffic case.

# Agradecimentos

Desejo expressar meus agradecimentos a todos que, de alguma forma, ajudaram na realização deste trabalho:

- Aos meus pais, familiares e amigos, pelo apoio e incentivo;
- Aos meus orientadores, prof. Dalton e prof. Nelson, pela amizade e orientação;
- À Faculdade de Engenharia Elétrica e de Computação da Unicamp, pela oportunidade de participar de seu programa de pós-graduação;
- À Fundação de Amparo à Pesquisa do Estado de São Paulo, pelo suporte a este projeto<sup>1</sup>.
- Ao colega César Augusto Viana Neto pela ajuda com o gerador de séries fBm e com o algoritmo do Balde Furado Fractal.

Finalmente, gostaria também de expressar agradecimento a todos que contribuíram com revisões e sugestões para a melhoria deste trabalho.

---

<sup>1</sup>Este projeto foi suportado pela Fundação de Amparo à Pesquisa do Estado de São Paulo sob o processo 00/00069-0.

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Tráfego auto-similar em redes</b>	<b>4</b>
2.1	Introdução . . . . .	4
2.2	Processos estocásticos auto-similares . . . . .	5
2.3	Processos auto-similares de segunda ordem . . . . .	8
2.3.1	Definição . . . . .	8
2.3.2	Propriedades dos processos auto-similares de segunda ordem . . . . .	11
2.4	Estimação do parâmetro de auto-similaridade . . . . .	11
2.4.1	Estimador de Whittle . . . . .	11
2.4.2	Gráfico variância-tempo . . . . .	12
2.4.3	Gráfico da estatística R/S . . . . .	13
2.4.4	Estimador baseado em <i>wavelets</i> . . . . .	14
2.5	Modelagem de tráfego auto-similar . . . . .	16
<b>3</b>	<b>O movimento Browniano fracionário</b>	<b>19</b>
3.1	Introdução . . . . .	19
3.2	Síntese do ruído Gaussiano fracionário . . . . .	20
3.3	O tráfego Browniano fracionário . . . . .	21
3.3.1	Armazenamento do tráfego Browniano fracionário . . . . .	22
3.3.2	Escala de Tempo Crítica . . . . .	25
3.4	O processo envelope do movimento Browniano fracionário . . . . .	26
3.4.1	Armazenamento do processo envelope do fBm . . . . .	27
3.4.2	Escala de tempo máxima . . . . .	28

<b>4</b>	<b>Policiamento de tráfego auto-similar</b>	<b>30</b>
4.1	Técnicas de policiamento de tráfego . . . . .	31
4.1.1	Técnicas de policiamento baseadas em janelas . . . . .	31
4.1.2	Algoritmo do Balde Furado . . . . .	32
4.1.3	Análise do algoritmo do Balde Furado . . . . .	32
4.2	Policiamento de tráfego auto-similar . . . . .	36
4.2.1	Policiamento utilizando o algoritmo do balde furado . . . . .	37
4.2.2	Algoritmo do Balde Furado Fractal . . . . .	39
4.2.3	Uso de <i>buffers</i> com o algoritmo do Balde Furado Fractal . . . . .	39
<b>5</b>	<b>Análise da disciplina de serviço GPS</b>	<b>42</b>
5.1	A disciplina de serviço GPS . . . . .	44
5.1.1	Aproximações para a disciplina GPS . . . . .	44
5.2	Análise de desempenho da disciplina GPS . . . . .	45
5.2.1	Sessões localmente estáveis . . . . .	45
5.2.2	Estabilidade de um servidor GPS . . . . .	46
5.2.3	Análise de desempenho de servidores GPS pela curva de serviço . . . . .	47
5.3	Exemplo numérico . . . . .	56
<b>6</b>	<b>Análise de desempenho de redes de servidores GPS</b>	<b>60</b>
6.1	Modelo de rede . . . . .	60
6.2	Estabilidade de redes de servidores GPS sob tráfego auto-similar . . . . .	61
6.2.1	Sessões localmente estáveis . . . . .	61
6.2.2	Cálculo do backlog e do atraso fim-a-fim das sessões em uma rede de servidores GPS . . . . .	65
6.3	Exemplo numérico . . . . .	70
6.4	Considerações finais . . . . .	74
6.4.1	Atraso de propagação nos enlaces . . . . .	74
6.4.2	Limitantes para variação de atraso . . . . .	75
<b>7</b>	<b>Conclusões</b>	<b>76</b>

# Lista de Figuras

2.1	Comparação entre os incrementos de um processo estocástico auto-similar e de um processo estocástico comum, sob o ponto de vista da preservação das propriedades estatísticas em diferentes escalas de agregação (1, 10 e 1000 amostras). . . . .	6
2.2	Comportamento da função de autocorrelação de um processo auto-similar de segunda ordem em função do parâmetro de auto-similaridade $H$ . . . . .	9
2.3	Estimação do parâmetro de auto-similaridade $H$ utilizando o gráfico variância-tempo. . . . .	13
2.4	Estimação do parâmetro de auto-similaridade $H$ utilizando o gráfico da estatística R/S. . . . .	14
2.5	Estimação do parâmetro de auto-similaridade $H$ utilizando o método baseado em <i>wavelets</i> . . . . .	16
3.1	Aproximação Weibull de $\log_{10} \mathbb{P}\{\mathbf{Q}^* > x\}$ para $\rho = 2279kb/s$ , $\sigma = 262.8kb \cdot s$ , $g = 4.8Mb/s$ e diversos valores de $H$ [30]. . . . .	24
4.1	O algoritmo do Balde Furado e as possibilidades de tratamento da parcela de tráfego correspondente a violações: (a) descarte; (b) marcação para que seja prioritária no descarte; e (c) armazenamento para posterior transmissão. . . . .	33
4.2	Restrição imposta ao tráfego pelo algoritmo do Balde Furado [32]. . . . .	34
4.3	Sessão exatamente conforme ao envelope do algoritmo do Balde Furado a partir do instante $\tau$ [32]. . . . .	36
4.4	Uso de retas tangentes ao processo $\hat{A}_i(t)$ no ajuste do algoritmo do Balde Furado. . . . .	38
5.1	Modelo básico para um servidor GPS. . . . .	45
5.2	Construção da função $\hat{S}_i(t)$ e definição de $Q_i^*$ e $D_i^*$ para $i = 3$ . . . . .	52
5.3	Resultados obtidos para a Sessão 1, sob tráfego exatamente conforme desde o instante zero. . . . .	57
5.4	Resultados obtidos para a Sessão 2, sob tráfego exatamente conforme desde o instante zero. . . . .	57
5.5	Resultados obtidos para a Sessão 3, sob tráfego exatamente conforme desde o instante zero. . . . .	58
5.6	Resultados obtidos para a Sessão 1, quando o tráfego das sessões é representado por seqüências fBm. . . . .	58
5.7	Resultados obtidos para a Sessão 2, quando o tráfego das sessões é representado por seqüências fBm. . . . .	59



5.8	Resultados obtidos para a Sessão 3, quando o tráfego das sessões é representado por seqüências fBm. . . .	59
6.1	Exemplo de construção da curva $U_i(t)$ considerando que $P(i)$ é composto por dois servidores. . .	68
6.2	Rede utilizada no exemplo. . . . .	70
6.3	Comparação entre o limitante do atraso fim-a-fim da Sessão 1 e os valores obtidos por simulação, sob tráfego exatamente conforme. . . . .	71
6.4	Comparação entre o limitante do atraso fim-a-fim da Sessão 2 e os valores obtidos por simulação, sob tráfego exatamente conforme. . . . .	72
6.5	Comparação entre o limitante do atraso fim-a-fim da Sessão 3 e os valores obtidos por simulação, sob tráfego exatamente conforme. . . . .	72
6.6	Comparação entre o limitante do atraso fim-a-fim da Sessão 1 e os valores obtidos por simulação, utilizando seqüências fBm. . . . .	73
6.7	Comparação entre o limitante do atraso fim-a-fim da Sessão 2 e os valores obtidos por simulação, utilizando seqüências fBm. . . . .	73
6.8	Comparação entre o limitante do atraso fim-a-fim da Sessão 3 e os valores obtidos por simulação, utilizando seqüências fBm. . . . .	74

# Capítulo 1

## Introdução

Nos últimos anos, os sistemas de comunicação se desenvolveram de forma extremamente rápida, acompanhando a evolução das aplicações em direção à convergência entre o transporte de dados e de informação multimídia, na denominada Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL).

As aplicações destinadas a RDSI-FL apresentam rígidas exigências de desempenho em termos de vazão, atraso, variação de atraso e taxa de perdas, que as tecnologias de comutação tradicionais não são capazes de suportar. Existem, porém, diversas propostas para o suporte ao modelo de serviço garantido exigido pela RDSI-FL. O Modo de Transferência Assíncrono (*Asynchronous Transfer Mode*, ATM) foi a proposta inicial para a sua implementação; no entanto, seu uso foi bastante limitado devido à sua excessiva complexidade, às dificuldades na sua padronização e, principalmente, à sua compatibilidade apenas parcial com o IP (*Internet Protocol*), que é o protocolo de rede mais difundido atualmente. Embora o IP não seja capaz de suportar o fornecimento de garantias de desempenho, surgiram também diversas propostas visando a incorporar-lhe os mecanismos necessários a este propósito. Dentre tais propostas destaca-se, por exemplo, o MPLS (*Multilayer Protocol Label Switching*).

Uma característica comum a todas estas propostas é o uso de mecanismos de controle de tráfego para o suporte do modelo de serviço garantido, dentre os quais destacam-se o controle de admissão, o policiamento do usuário e o escalonamento de pacotes.

O controle de admissão visa a restringir o acesso à rede apenas aos usuários que possam ser atendidos sem que haja degradação da qualidade do serviço prestado. Para isso, a admissão de um novo usuário é sempre precedida de uma negociação, na qual ele fornece seus requisitos de desempenho e as informações necessárias à caracterização do seu tráfego. O novo usuário é admitido somente se a rede for capaz de atendê-lo sem prejuízo aos demais. Neste caso, é estabelecido um compromisso de prestação de serviço: a rede se compromete a atender os requisitos de desempenho do usuário, desde que o seu tráfego se mantenha em conformidade com as especificações fornecidas na

fase de negociação. Cabe ao mecanismo de policiamento impedir que o usuário viole este compromisso, detectando e tratando desvios que poderiam levar à degradação da qualidade do serviço prestado aos demais usuários.

A função do escalonamento é controlar a interação entre os pacotes dos diversos usuários da rede, estabelecendo a ordem em que são atendidos ou descartados pelo comutador. Em particular, esta ordem é estabelecida segundo um critério denominado política de escalonamento ou disciplina de serviço.

Pode-se dizer que a disciplina de serviço desempenha um papel fundamental no controle de tráfego, pois cabe a ela gerenciar três recursos [47]: a largura de faixa (quais pacotes são transmitidos), a prontidão (quando estes pacotes são transmitidos) e as filas de espera no comutador (quais pacotes são descartados). Estes recursos, por sua vez, afetam diretamente a vazão, o atraso, a variação de atraso e a taxa de perdas de pacotes, que são os índices de desempenho normalmente especificados pelos usuários.

Diante da sua importância no contexto do modelo de serviço garantido, é importante a escolha das disciplinas de serviço a serem empregadas nos comutadores. Para que seja considerada adequada, uma disciplina de serviço deve apresentar as seguintes características [47]:

- **Eficiência:** Uma disciplina de serviço é mais eficiente que outra se ela suporta um maior volume de tráfego fornecendo as mesmas garantias de desempenho para cada usuário, ou seja, ela permite o uso mais eficiente dos recursos da rede;
- **Proteção:** Ao fornecer garantias de serviço aos usuários, a rede se compromete a proteger seus fluxos de dados de flutuações de tráfego na rede, de excessos de tráfego gerado por outros usuários e do tráfego do tipo “melhor esforço”. Embora cada um destes fatores seja tratado separadamente por outros mecanismos, é fundamental que a disciplina de serviço viabilize o suporte às garantias de serviço fornecidas aos usuários;
- **Flexibilidade:** De modo a suportar aplicações diversificadas, a rede deve estar apta a tratar diferentes perfis de tráfego e requisitos de desempenho, o que exige também o emprego de disciplinas de serviço flexíveis;
- **Simplicidade:** Visto que ela opera na menor escala de tempo do controle de tráfego (escala de pacotes), a disciplina de serviço deve ser simples para que sua análise de desempenho e sua implementação computacional sejam viáveis.

## Análise de desempenho de disciplinas de serviço

O objetivo da análise de desempenho de disciplinas de serviço é estabelecer relações entre os parâmetros de tráfego e os índices de desempenho de cada conexão — relações estas que podem ser utilizadas pelos mecanismos de controle de admissão para determinar se uma nova conexão pode ser aceita e qual parcela da capacidade de transmissão deve ser a ela destinada.

Neste trabalho, tratar-se-á do problema da análise da disciplina de serviço denominada *Generalized Processor Sharing* (GPS), que apresenta diversas características interessantes no contexto das redes de serviço garantido, conforme já foi demonstrado por diversos autores. Um resumo dos principais trabalhos que tratam da análise de desempenho do GPS pode ser encontrado no Capítulo 5 desta dissertação. Pode-se, no entanto, destacar os estudos de Parekh e Gallager [32, 33], que tratam da análise de desempenho do GPS quando o tráfego dos usuários é policiado pelo algoritmo do Balde Furado. Neste caso é demonstrado que, sob certas condições, limitantes para o atraso e para o *backlog* podem ser obtidos tanto para um servidor GPS isolado como para uma rede cujos nós utilizem esta disciplina de serviço.

A recente constatação do caráter auto-similar do tráfego multimídia tornou ainda mais importante o problema da análise de desempenho, visto que a auto-similaridade pode degradar significativamente o desempenho da rede [30, 36]. No entanto, a grande maioria dos trabalhos voltados à análise da disciplina GPS não levam a auto-similaridade em consideração, e mesmo aqueles que são baseados em modelos auto-similares apresentam resultados apenas de caráter preliminar.

O objetivo deste trabalho é, então, analisar o comportamento de um servidor GPS alimentado por tráfego auto-similar. Assume-se que o tráfego dos usuários é policiado pelo algoritmo do Balde Furado Fractal, que é um mecanismo de policiamento adequado a este tipo de tráfego [14, 26]. É obtido um algoritmo que permite caracterizar o desempenho do servidor, em particular quanto ao atraso e ao *backlog* de cada sessão. Em seguida, estes resultados são estendidos para redes de servidores GPS com topologia arbitrária, cuja estabilidade também é analisada.

Os demais capítulos têm a seguinte organização. No Capítulo 2, é discutido o tráfego auto-similar e seus efeitos. No Capítulo 3, são apresentados o tráfego Browniano fracionário e o processo envelope do movimento Browniano fracionário. O Capítulo 4 trata dos mecanismos de policiamento de tráfego, mostrando a ineficácia do algoritmo do Balde Furado no policiamento de tráfego auto-similar, e apresentando o algoritmo do Balde Furado Fractal. No Capítulo 5, a disciplina GPS é apresentada, sendo efetuada a análise de desempenho de um servidor GPS isolado. No Capítulo 6, uma rede de servidores GPS com topologia arbitrária é analisada, com foco nas questões de estabilidade e na provisão de limitantes para o *backlog*, para o atraso fim-a-fim e para a variação do atraso fim-a-fim. Finalmente, no Capítulo 7 são apresentadas as conclusões e possíveis extensões deste trabalho.

## Capítulo 2

# Tráfego auto-similar em redes

### 2.1 Introdução

A auto-similaridade é um conceito que foi introduzido por Mandelbrot [23] para caracterizar a preservação, em relação à escala espacial ou temporal de observação, de certas propriedades de uma classe de objetos denominados fractais. No contexto das redes de comunicação, a auto-similaridade ganhou importância após o estudo de Leland, Taqqu, Willinger e Wilson [22], que constataram a natureza auto-similar do tráfego em redes *ethernet*. Este estudo, baseado na análise do tráfego do *Bellcore Morristown Research and Engineering Center*, mostra que o tráfego *ethernet* é composto pela alternância de períodos de surtos e de suavidade, o que é observado em diversas escalas de agregação. Este comportamento não pode ser adequadamente reproduzido pelos modelos baseados em processos estocásticos markovianos, até então utilizados para dimensionamento e análise de desempenho de redes.

Pode-se dizer que os resultados apresentados por Leland et al. marcaram o início de uma nova linha de pesquisa que, hoje, aponta a auto-similaridade no tráfego gerado por transmissão de vídeo a taxa variável [2, 15] e por aplicações de rede geograficamente distribuídas, tais como aplicações cliente-servidor, transferência de arquivos [39], aplicações *World-Wide Web* [8], dentre outras.

As causas da presença de auto-similaridade no tráfego ainda não foram determinadas de modo conclusivo. Inicialmente, ela era considerada consequência do comportamento de fontes isoladas cujo tráfego é reconhecidamente auto-similar (e.g. transmissão de vídeo comprimido). No entanto, Willinger, Taqqu, Sherman e Wilson [44] demonstraram que um fluxo de tráfego auto-similar pode ser obtido pela superposição de muitas fontes *on/off* identicamente distribuídas, desde que os períodos de silêncio e de atividade sejam representados por variáveis aleatórias com distribuição de cauda longa, i.e., que podem assumir valores elevados com probabilidade não-negligenciável (efeito Noé). Este resultado é comprovado por outros estudos, que mostram que a agregação do tráfego de diferentes aplicações envolvendo transferência de arquivos (e.g. aplicações cliente-servidor e *World-Wide*

Web) pode resultar em um fluxo auto-similar se o tamanho dos arquivos segue uma distribuição subexponencial [8, 34, 36].

Do ponto de vista da análise e do dimensionamento de redes, ainda não há consenso em relação à relevância da auto-similaridade. Alguns autores argumentam que, sob certas condições, ela tem pouca influência no comportamento do tráfego [17]. No entanto, a grande maioria dos estudos mostra que a auto-similaridade pode degradar o desempenho das redes de forma significativa. Neste caso, a incidência de surtos de tráfego tende a ser mais alta, e o decaimento da distribuição probabilística da ocupação dos *buffers* dos nós de comutação tende a ser muito lento [28, 34, 35, 36, 39]. Isto significa que:

- a taxa de perda de pacotes por transbordo cresce rapidamente com o grau de auto-similaridade, e não é possível obter significativa redução com o aumento dos *buffers* nos comutadores;
- a alta incidência de surtos no tráfego auto-similar reduz a eficiência da multiplexação estatística no aumento do uso efetivo da capacidade de transmissão;
- os atrasos de transferência tendem a aumentar significativamente, já que a ocupação média dos *buffers* cresce rapidamente com o grau de auto-similaridade do tráfego.

## 2.2 Processos estocásticos auto-similares

Os processos estocásticos auto-similares são particularmente interessantes na modelagem de tráfego, pois permitem representar a auto-similaridade por meio de um único parâmetro. A Figura 2.1(a) apresenta a realização dos incrementos de um processo estocástico auto-similar sob diferentes escalas de agregação (1, 10 e 1000 amostras). Embora as curvas não sejam exatamente iguais, pode-se constatar que as propriedades estatísticas são preservadas com a agregação: a alternância de períodos de surtos e de suavidade é preservada em todas as escalas. Como consequência, o processo auto-similar mantém suas propriedades estatísticas em relação à escala de tempo de observação. Tal característica não se verifica em um processo estocástico que não seja auto-similar, conforme ilustra a Figura 2.1(b): neste caso, a agregação resulta em um ruído gaussiano branco, o que indica que as propriedades estatísticas do processo não são preservadas.

Matematicamente, pode-se definir um processo estocástico auto-similar da seguinte forma:

**Definição 2.1.** (Processos estocásticos auto-similares) Seja um processo estocástico  $\mathbf{Y}(t)$ ,  $t \in \mathbb{R}$ . Este processo é dito auto-similar com parâmetro de auto-similaridade  $H \in (0; 1)$  se, para todo  $a > 0$  e  $t \geq 0$ , os processos  $\mathbf{Y}(t)$  e  $a^{-H}\mathbf{Y}(at)$  são identicamente distribuídos, i.e.

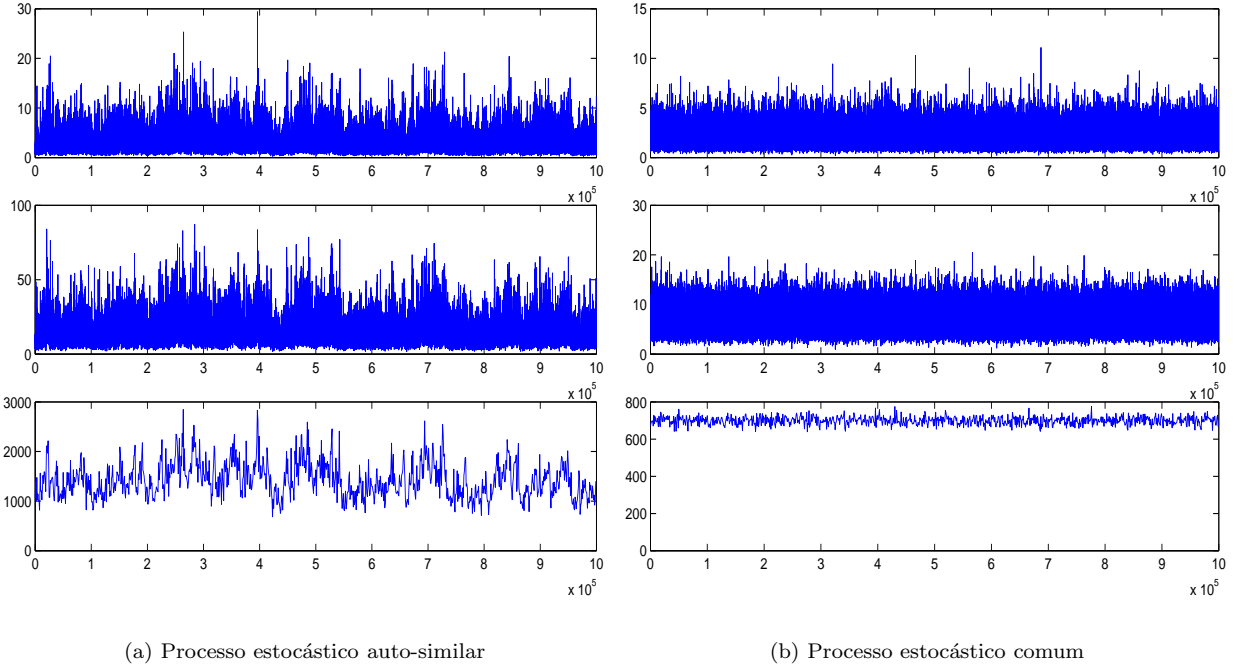


Figura 2.1: Comparação entre os incrementos de um processo estocástico auto-similar e de um processo estocástico comum, sob o ponto de vista da preservação das propriedades estatísticas em diferentes escalas de agregação (1, 10 e 1000 amostras).

$$\mathbf{Y}(t) \stackrel{d}{=} a^{-H} \mathbf{Y}(at) \quad (2.1)$$

onde o símbolo  $\stackrel{d}{=}$  denota a igualdade em distribuição.

**Lema 2.2.** Para um processo  $\mathbf{Y}(t)$  que seja auto-similar,

$$\mathbf{Y}(t) \stackrel{d}{=} t^H \mathbf{Y}(1)$$

o que pode ser verificado substituindo, na relação (2.1),  $a = t^{-1}$ .

O parâmetro de auto-similaridade  $H$  é denominado parâmetro de Hurst, em homenagem a H. E. Hurst [19]. Um processo  $\mathbf{Y}(t)$  que seja auto-similar, segundo a Definição 2.1, é obrigatoriamente não-estacionário<sup>1</sup>. No contexto da modelagem de tráfego, este processo representa o volume de tráfego acumulado até o instante  $t$  sendo, portanto, denominado processo de acumulação. A este processo pode ser associado um outro, denominado

---

<sup>1</sup>A menos que seja degenerado (i.e.  $\mathbf{Y}(t) = 0, \forall t \in \mathbb{R}$ ).

processo de incrementos, que representa o volume de tráfego acumulado no intervalo  $[t_o; t_o + t]$ . O conceito de auto-similaridade pode também ser definido para processos de incrementos [24]:

**Definição 2.3.** O processo de incrementos  $\mathbf{X}(t)$ ,  $t \in \mathbb{R}$ , associado a um dado processo  $\mathbf{Y}(t)$ , é dito auto-similar com parâmetro de auto-similaridade  $H \in (0; 1)$  se, para todo  $a > 0$  e  $t_o \geq 0$ :

$$\mathbf{X}(t) = \mathbf{Y}(t_o + t) - \mathbf{Y}(t_o) \stackrel{d}{=} a^{-H} [\mathbf{Y}(t_o + at) - \mathbf{Y}(t_o)]$$

É possível demonstrar que, se o processo  $\mathbf{Y}(t)$  é auto-similar, o correspondente processo de incrementos  $\mathbf{X}(t)$  também é auto-similar. No contexto da modelagem de tráfego, é interessante considerar apenas os processos  $\mathbf{Y}(t)$  cujo processo de incrementos  $\mathbf{X}(t)$  seja estacionário, ao menos no sentido amplo. Desta forma, o processo  $\mathbf{X}(t)$  pode ser discretizado e utilizado para representar o volume de tráfego em um instante  $t \in \mathbb{Z}$ . Neste sentido, seja  $\mathbf{Y}(t)$  um processo auto-similar com incrementos estacionários, para o qual  $\mathbb{E}\{\mathbf{Y}(t)\} = 0$ . A partir do correspondente processo de incrementos em tempo discreto,  $\mathbf{X}(t)$ ,  $t \in \mathbb{Z}$ , podem ser obtidos processos agregados  $\mathbf{X}^{(m)}(t)$ ,  $t \in \mathbb{Z}$ , definidos como a média amostral do processo original  $\mathbf{X}(t)$  em blocos não sobrepostos de tamanho  $m$ , i.e.

$$\mathbf{X}^{(m)}(t) = \frac{1}{m} \sum_{i=m(t-1)+1}^{mt} \mathbf{X}(i)$$

Dado que o processo  $\mathbf{X}(t)$  é estacionário,

$$\begin{aligned} \mathbf{X}^{(m)} &\stackrel{d}{=} \frac{1}{m} \sum_{i=1}^m \mathbf{X}(i) \\ &\stackrel{d}{=} \frac{1}{m} [\mathbf{Y}(m) - \mathbf{Y}(0)] \\ &\stackrel{d}{=} m^{H-1} [\mathbf{Y}(1) - \mathbf{Y}(0)] \\ &\stackrel{d}{=} m^{H-1} \mathbf{X} \end{aligned}$$

Assim, se o processo  $\mathbf{Y}(t)$  é auto-similar com incrementos estacionários, então o correspondente processo de incrementos em tempo discreto satisfaz uma relação semelhante àquela utilizada para definir a auto-similaridade em tempo contínuo, e a seguinte definição pode ser estabelecida:



**Definição 2.4.** (Processos estocásticos discretos e auto-similares) Seja um processo estocástico  $\mathbf{X}(t)$ ,  $t \in \mathbb{Z}$ . Este processo é dito auto-similar com parâmetro de auto-similaridade  $H \in (0; 1)$  se, para todo  $m > 0$  e  $t \geq 0$ , os processos  $\mathbf{X}(t)$  e  $m^{1-H} \mathbf{X}^{(m)}(t)$  são identicamente distribuídos, i.e.

$$\mathbf{X}(t) \stackrel{d}{=} m^{1-H} \mathbf{X}^{(m)}(t) \quad (2.2)$$

Os processos estocásticos que satisfazem (2.2) são denominados exatamente auto-similares, visto que a sua distribuição se mantém invariável em todas as escalas de agregação. No entanto, é possível que a auto-similaridade se manifeste somente em escalas de agregação maiores, o que torna interessante introduzir também a seguinte definição:

**Definição 2.5.** (Processos estocásticos discretos e assintoticamente auto-similares) Seja um processo estocástico  $\mathbf{X}(t)$ ,  $t \in \mathbb{Z}$ . Este processo é dito assintoticamente auto-similar com parâmetro de auto-similaridade  $H \in (0; 1)$  se, para todo  $t \geq 0$ ,

$$\lim_{m \rightarrow \infty} m^{1-H} \mathbf{X}^{(m)}(t) \stackrel{d}{=} \mathbf{X}(t) \quad (2.3)$$

## 2.3 Processos auto-similares de segunda ordem

### 2.3.1 Definição

Em muitas aplicações, apenas algumas propriedades estatísticas dos processos são consideradas relevantes. No caso das redes, por exemplo, o tráfego é normalmente caracterizado em termos de estatísticas de segunda ordem, de modo que é suficiente considerar a manifestação da auto-similaridade apenas nestas estatísticas. Isto permite relaxar as Definições 2.4 e 2.5 e introduzir a seguinte definição [36]:

**Definição 2.6.** (Processos estocásticos discretos e exatamente auto-similares de segunda ordem) Seja um processo estocástico  $\mathbf{X}(t)$ ,  $t \in \mathbb{Z}$ . Este processo é denominado exatamente auto-similar de segunda ordem com parâmetro  $H \in [1/2; 1)$  se for possível exprimir sua autocorrelação como:

$$\mathcal{R}_{\mathbf{X}^{(m)}}(\tau) = \mathcal{R}_{\mathbf{X}}(\tau) = \frac{1}{2} [(\tau + 1)^{2H} - 2\tau^{2H} + (\tau - 1)^{2H}] , \forall m \geq 1 \quad (2.4)$$

**Definição 2.7.** (Processos estocásticos discretos e assintoticamente auto-similares de segunda ordem) Seja um processo estocástico  $\mathbf{X}(t)$ ,  $t \in \mathbb{Z}$ . Este processo é denominado assintoticamente auto-similar de segunda ordem com parâmetro  $H \in [1/2; 1)$  se for possível exprimir sua autocorrelação como:

$$\lim_{m \rightarrow \infty} \mathcal{R}_{\mathbf{X}(m)}(\tau) = \mathcal{R}_{\mathbf{X}}(\tau) = \frac{1}{2} [(\tau + 1)^{2H} - 2\tau^{2H} + (\tau - 1)^{2H}] \quad (2.5)$$

*Comentário.* No conceito de auto-similaridade de segunda ordem, o parâmetro  $H$  é normalmente restrito ao intervalo  $[1/2; 1)$  por ser esta a região de interesse prático — a condição  $H \in (0; 1/2)$  é dificilmente encontrada na prática visto que, neste caso,  $\sum_{\tau} \mathcal{R}_{\mathbf{X}}(\tau) = 0$ .

A principal característica dos processos auto-similares de segunda ordem é, portanto, o caráter não-degenerativo da função de autocorrelação dos seus processos agregados. A Figura 2.2 ilustra o comportamento da função de auto-correlação  $\mathcal{R}_{\mathbf{X}}(\tau)$  para diversos valores de  $H$ .

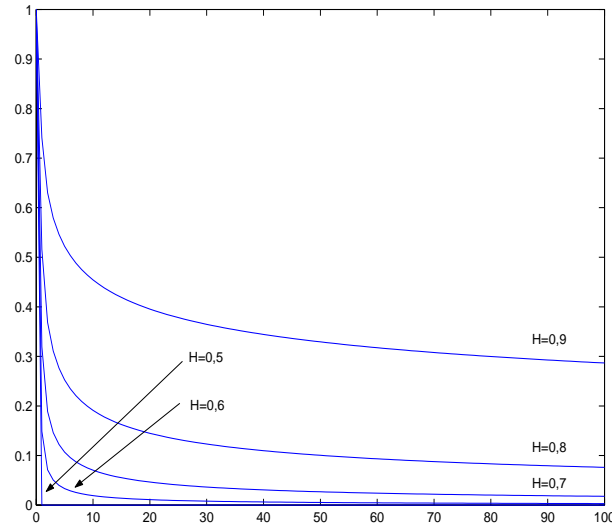


Figura 2.2: Comportamento da função de autocorrelação de um processo auto-similar de segunda ordem em função do parâmetro de auto-similaridade  $H$ .

Nota-se que, para valores baixos de  $H$ , valores expressivos de autocorrelação são encontrados apenas para pequenos valores de  $\tau$ . Em particular para  $H = 1/2$ , a função de autocorrelação é equivalente àquela do ruído gaussiano branco sendo, portanto, somável (i.e.  $\sum_{\tau} \mathcal{R}_{\mathbf{X}}(\tau) < \infty$ ). Neste caso, diz-se que o processo  $\mathbf{X}(t)$  apresenta dependência de curta duração (*short-range dependence*, SRD).

Por outro lado, para  $H > 1/2$ , a relação (2.4) pode ser aproximada por [36]:

$$\mathcal{R}_{\mathbf{X}}(\tau) \simeq H(2H - 1)\tau^{2H-2}, \tau \rightarrow \infty$$

o que significa que, do ponto de vista assintótico,  $\mathcal{R}_{\mathbf{X}}(\tau)$  se comporta como  $c\tau^{-\beta}$ ,  $0 < \beta < 1$ . Isto significa que a função de autocorrelação do processo  $\mathbf{X}(t)$  apresenta decaimento hiperbólico e, portanto, não é somável (i.e.  $\sum_{\tau} \mathcal{R}_{\mathbf{X}}(\tau) \rightarrow \infty$ ). Além disso, pode-se dizer que, nas proximidades da origem, a sua densidade espectral de potência  $S_{\mathbf{X}}(\omega)$  se comporta como  $c\omega^{-\gamma}$  ( $0 < \gamma < 1$ ), de forma que  $\lim_{\omega \rightarrow 0} S_{\mathbf{X}}(\omega) \rightarrow \infty$ . Tal comportamento não se verifica no caso de  $H = 1/2$ , para o qual a densidade espectral de potência é limitada na origem. Assim, diz-se que processos auto-similares com  $H > 1/2$  apresentam dependência de longa duração (*long-range dependence*, LRD), cujo principal efeito é a persistência do comportamento do processo no domínio do tempo.

**Definição 2.8.** (Processos estocásticos estacionários com dependência de longa duração) Um processo estocástico estacionário  $\mathbf{X}(t)$  apresenta dependência de longa duração se a sua função de autocorrelação apresenta decaimento hiperbólico, i.e.,  $\mathcal{R}_{\mathbf{X}}(\tau)$  comporta-se assintoticamente como  $c\tau^{-\beta}$ ,  $0 < \beta < 1$ .

Aos processos com dependência de longa duração são geralmente associadas variáveis aleatórias de distribuição de cauda longa, cuja função de distribuição de probabilidade decai de forma hiperbólica. Matematicamente, uma função de distribuição de cauda longa é definida como [16]:

**Definição 2.9.** (Função de distribuição de cauda longa) Seja  $F(x)$  uma função distribuição de probabilidade definida no intervalo  $(0; \infty)$ , tal que  $F(x) < 1$ ,  $\forall x > 0$ . A função  $F(x)$  é dita uma distribuição de cauda longa se

$$\lim_{x \rightarrow \infty} \frac{\overline{F}(x - y)}{\overline{F}(x)} = 1 \quad \forall y > 0$$

sendo  $\overline{F}(x) = 1 - F(x)$ .

É possível demonstrar que uma variável aleatória  $\mathbf{X}$  tem distribuição de cauda longa se, para  $x \rightarrow \infty$ ,  $\mathbb{P}\{\mathbf{X} > x\} \sim x^{-\alpha}$ ,  $0 < \alpha < 2$  [36]. Devido a sua relação com as distribuições de cauda longa, pode-se dizer que os processos com dependência de longa duração atingem valores elevados com probabilidade não-negligenciável. Em conjunto com o comportamento persistente no domínio do tempo, tal fato justifica os efeitos danosos do tráfego auto-similar, em particular para  $H > 1/2$ . Por outro lado, a dependência de longa duração torna o processo passível de predição [36], o que pode ser utilizado no desenvolvimento de mecanismos de controle preditivo de tráfego [18].

### 2.3.2 Propriedades dos processos auto-similares de segunda ordem

Se um processo  $\mathbf{X}(t)$  é auto-similar de segunda ordem com  $H > 1/2$ , ele apresenta as seguintes propriedades:

- A variância da sua média temporal decresce mais lentamente que a recíproca do número de amostras utilizadas no seu cálculo. Isto significa que  $\text{Var} \{ \mathbf{X}^{(m)}(t) \} \sim am^{2-2H}$  para  $m \rightarrow \infty$ , sendo  $a$  uma constante finita e independente de  $m$ .
- A sua função de autocorrelação tem decaimento hiperbólico e, portanto, sua soma diverge ao infinito. Isto significa que :
  - O processo  $\mathbf{X}(t)$  apresenta dependência de longa duração;
  - A densidade espectral de potência  $S_{\mathbf{X}}(\omega)$  tende ao infinito para  $\omega \rightarrow 0$ . De fato, nas proximidades da origem  $S_{\mathbf{X}}(\omega) \sim c\omega^{1-2H}$ , sendo  $c$  uma constante finita.

Por outro lado, tais propriedades não são observadas em processos que não sejam auto-similares, ou em processos auto-similares com  $H = 1/2$ . Nestes casos:

- A variância da média temporal decresce linearmente com a recíproca do número de amostras utilizadas no seu cálculo, i.e.,  $\text{Var} \{ \mathbf{X}^{(m)}(t) \} \sim am^{-1}$  para  $m \rightarrow \infty$ , sendo  $a$  uma constante finita e independente de  $m$ .
- A função de autocorrelação tem decaimento exponencial sendo, portanto, somável. Desta forma estes processos apresentam dependência de curta duração e sua densidade espectral de potência é limitada na origem.

## 2.4 Estimação do parâmetro de auto-similaridade

Diversas técnicas podem ser utilizadas para estimar o parâmetro  $H$  de uma série temporal a partir das suas amostras. Pode-se citar o estimador de Whittle, os gráficos variância-tempo e de estatística R/S e o estimador baseado em *wavelets*.

### 2.4.1 Estimador de Whittle

Este método supõe que tanto a estrutura como os parâmetros do processo  $\mathbf{X}(t)$  são conhecidos, à exceção da variância e do parâmetro  $H$ . A densidade espectral de potência deste processo pode ser definida, a partir de amostras da sua função de autocorrelação, como:

$$S_{\mathbf{X}}(\omega, H) = \sum_{\tau} \mathcal{R}_{\mathbf{X}}(\tau) \exp(-i\omega\tau) \quad (2.6)$$

Define-se, então,  $S_{\mathbf{X}}^*(\omega, H)$  como uma forma normalizada de  $S_{\mathbf{X}}(\omega, H)$ , na qual a variância de  $\mathbf{X}(t)$  seria unitária, e ainda  $I_N(\omega)$  como o periodograma obtido a partir de  $N$  amostras do processo  $\mathbf{X}(t)$ :

$$I_N(\omega) = \frac{1}{2\pi N} \left| \sum_{k=1}^N X(k) \exp(i\omega k) \right|^2$$

O parâmetro de auto-similaridade pode, então, ser obtido resolvendo o seguinte problema de otimização:

$$H = \arg \min_H \int_{-\pi}^{\pi} \frac{I_N(\omega)}{S_{\mathbf{X}}^*(\omega, H)} d\omega \quad (2.7)$$

Se o processo  $\mathbf{X}(t)$  é reduzido a um conjunto de  $N$  amostras, a integral em (2.7) pode ser entendida como uma soma discreta para  $\omega = \frac{2\pi}{N}, \frac{4\pi}{N}, \dots, \frac{2(n-1)\pi}{N}$ . As definições utilizadas pelo estimador de Whittle permitem ainda obter diretamente a variância da estimativa,  $\sigma_H^2$ , definida como:

$$\sigma_H^2 = 4\pi \left[ \int_{-\pi}^{\pi} \left( \frac{\partial}{\partial H} \log S_{\mathbf{X}}^*(\omega, H) \right)^2 d\omega \right]^{-1}$$

Embora o estimador de Whittle permita uma análise bastante refinada dos dados, ele tem a desvantagem de ser um método paramétrico, podendo levar a resultados incorretos caso os dados não se ajustem à estrutura previamente assumida para o processo  $\mathbf{X}(t)$ .

### 2.4.2 Gráfico variância-tempo

Este método recorre à relação entre as variâncias do processo original e do processo obtido por agregação, dada por:

$$\text{Var} \{ \mathbf{X}(t) \} = m^{2-2H} \text{Var} \{ \mathbf{X}^{(m)}(t) \}$$

ou, em termos dos logaritmos:

$$\log \text{Var} \left\{ \mathbf{X}^{(m)}(t) \right\} = \log \text{Var} \left\{ \mathbf{X}(t) \right\} + (2H - 2) \log m$$

Portanto, para um conjunto de amostras de uma série temporal, a curva que representa a relação entre  $\log \text{Var} \left\{ \mathbf{X}^{(m)}(t) \right\}$  e  $\log m$  aproxima-se de uma reta de inclinação  $2H - 2$ . Na prática, esta curva pode ser obtida facilmente gerando os processos agregados para diversos valores de  $m$  e calculando sua variância, como mostra a Figura 2.3. A série temporal apresentada na Figura 2.3(a) corresponde a um ruído Gaussiano fracionário gerado artificialmente utilizando o método de Chi [5] para  $H = 0,8$ , e ao qual foi aplicada a transformação logarítmica proposta por Paxson [37].

Na Figura 2.3(b) é mostrado o gráfico variância-tempo correspondente ao processo  $\mathbf{X}(t)$ . Por meio deste gráfico é obtida uma estimativa de  $H$  igual a 0,737.

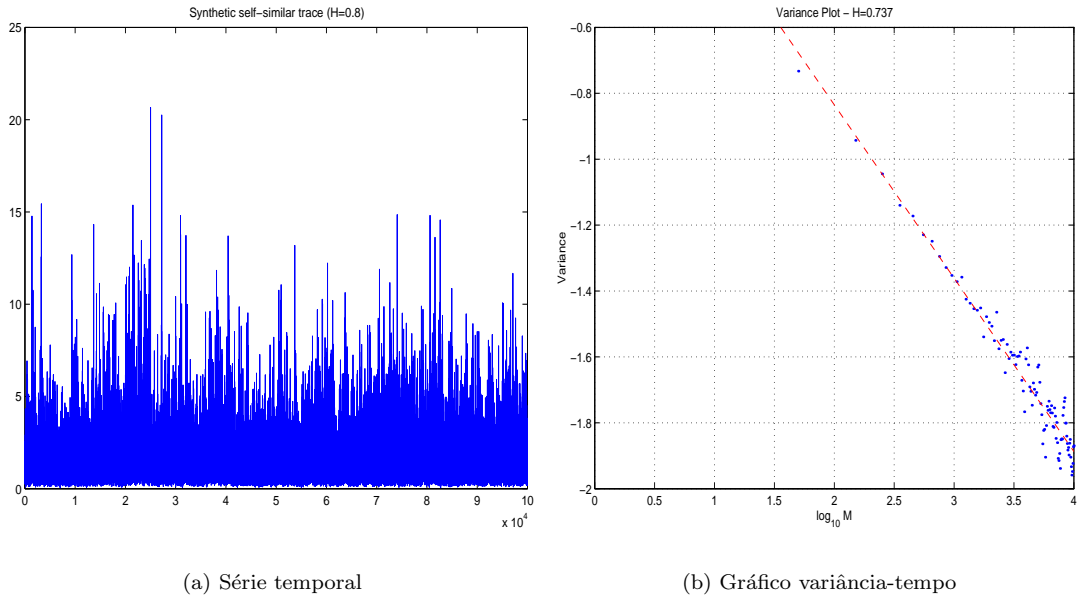


Figura 2.3: Estimação do parâmetro de auto-similaridade  $H$  utilizando o gráfico variância-tempo.

### 2.4.3 Gráfico da estatística R/S

Este método é baseado na definição da estatística R/S para processos estocásticos discretos, e sua principal vantagem é a independência com relação à distribuição marginal do processo em análise. A estatística R/S representa a razão entre a medida da faixa do processo  $\mathbf{X}(t)$  e o desvio padrão amostrado. Sendo  $X(t)$  a realização deste processo, esta razão é definida como:

$$R/S(N) = \frac{\max_{1 \leq j \leq N} \left\{ \sum_{l=1}^j [X(l) - M(N)] \right\} - \min_{1 \leq j \leq N} \left\{ \sum_{l=1}^j [X(l) - M(N)] \right\}}{\sqrt{\frac{1}{N} \sum_{l=1}^N [X(l) - M(N)]^2}}$$

sendo  $M(N) = \frac{1}{N} \sum_{l=1}^N X(l)$ . No caso de processos auto-similares, esta razão apresenta a seguinte característica [22]:

$$R/S(N) \sim N^H$$

Pode-se facilmente verificar que a curva que representa a relação entre  $\log(R/S)$  e  $\log(N)$  para diversos valores de  $N$  se aproxima de uma reta de inclinação  $H$ . A Figura 2.4 mostra a aplicação deste método na determinação do parâmetro  $H$  da mesma série temporal utilizada na Figura 2.3, sendo a estimativa de  $H$  igual a 0,792.

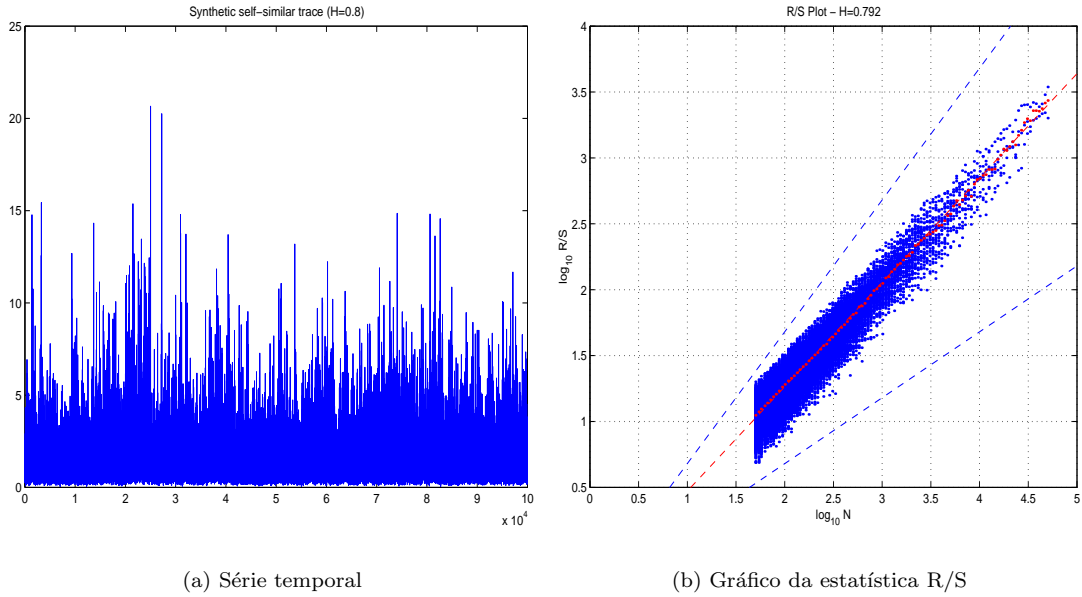


Figura 2.4: Estimação do parâmetro de auto-similaridade  $H$  utilizando o gráfico da estatística R/S.

#### 2.4.4 Estimador baseado em *wavelets*

O parâmetro de auto-similaridade de um processo pode ser estimado utilizando técnicas baseadas na transformada de *wavelets* [1, 18]. Seja  $\mathbf{Y}(t)$  um processo auto-similar segundo a Definição 2.1. O desenvolvimento em *wavelets* deste processo até a escala  $J$  é dado por:

$$\mathbf{Y}(t) = 2^{-J/2} \sum_{n=-\infty}^{\infty} \mathbf{a}_J[n] \phi(2^{-J}t - n) + \sum_{j=1}^J 2^{-j/2} \sum_{n=-\infty}^{\infty} \mathbf{d}_j[n] \psi(2^{-j}t - n)$$

onde  $\psi(t)$  é a função *wavelet* utilizada no desenvolvimento, e  $\phi(t)$  é a correspondente função de escala. Para uma decomposição ortonormal,

$$\begin{aligned} \mathbf{d}_j[n] &= 2^{-j/2} \int_{-\infty}^{\infty} \mathbf{Y}(t) \psi(2^{-j}t - n) dt \\ \mathbf{a}_J[n] &= 2^{-J/2} \int_{-\infty}^{\infty} \mathbf{Y}(t) \phi(2^{-J}t - n) dt \end{aligned}$$

onde  $j, n \in \mathbb{Z}$ . As seqüências  $\mathbf{a}_J[n]$  e  $\mathbf{d}_j[n]$  são denominadas, respectivamente, coeficientes de aproximação e coeficientes de detalhamento da transformada. É possível demonstrar que a variância de  $\mathbf{d}_j[n]$  é dada por [18]:

$$\text{Var} \{ \mathbf{d}_j[n] \} = k 2^{j(2H+1)} V_\psi(H)$$

A função  $V_\psi(H)$  depende da *wavelet* utilizada na decomposição. Para o caso da *wavelet* de Haar:

$$V_\psi(H) = \frac{1 - 2^{-2H}}{(H+1)(2H+1)}$$

De forma que:

$$\log_2 \text{Var} \{ \mathbf{d}_j[n] \} = (2H+1)j + \left[ 1 + \log_2 k + \log_2 \left( \frac{V_\psi(H)}{2} \right) \right]$$

A curva que representa a relação entre  $\log_2 \text{Var} \{ \mathbf{d}_j[n] \}$  e a escala  $j$  é, portanto, uma reta de inclinação  $2H-1$ . Se o processo  $\mathbf{Y}(t)$  for gaussiano e apresentar incrementos estacionários (i.e., processo do movimento Browniano fracionário, a ser discutido no Capítulo 3), então  $k = \text{Var} \{ \mathbf{Y}(t) \} / 2$ , e a variância também pode ser estimada utilizando este método — embora seja necessário efetuar a ponderação de  $\log_2 \text{Var} \{ \mathbf{d}_j[n] \}$  em função de  $j$  para reduzir o erro de estimação [18].



As principais vantagens do método baseado em *wavelets* é a sua eficiência e a capacidade de estimar o parâmetro  $H$  sem polarização. A Figura 2.5 mostra a aplicação deste método na determinação do parâmetro  $H$  da mesma série temporal utilizada na Figura 2.3, sendo a estimativa de  $H$  igual a 0,783.

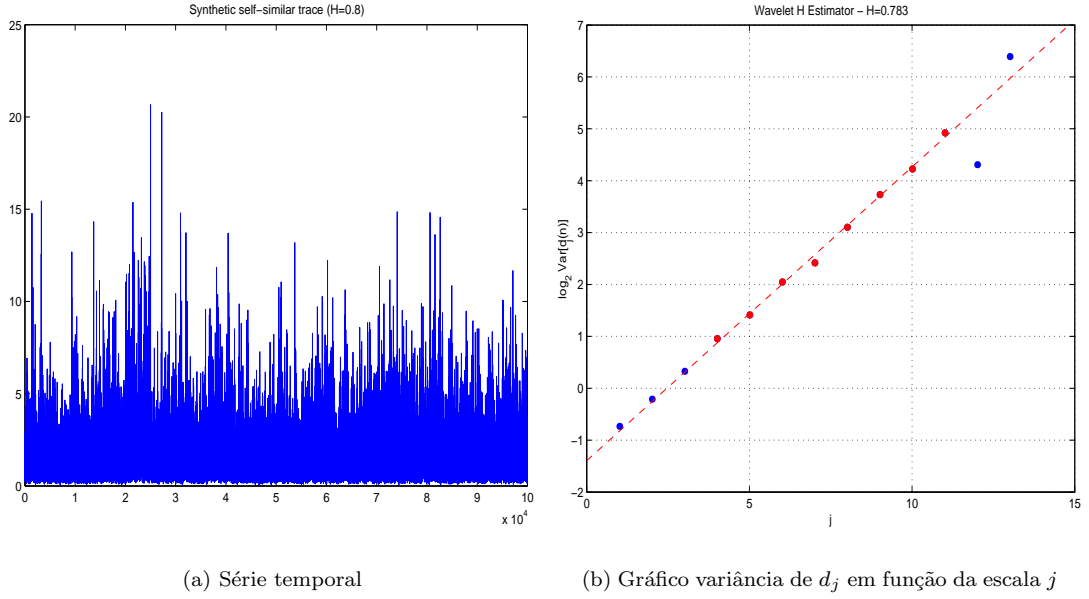


Figura 2.5: Estimação do parâmetro de auto-similaridade  $H$  utilizando o método baseado em *wavelets*.

## 2.5 Modelagem de tráfego auto-similar

Durante muito tempo, os modelos de tráfego utilizados na análise de redes foram baseados em processos markovianos. No entanto, tais modelos não são capazes de representar o tráfego auto-similar de maneira adequada, sobretudo quando ele apresenta também dependência de longa duração. Neste sentido, diversos modelos foram propostos, dentre os quais podem ser destacados:

- **Modelo *on/off* de cauda longa**

Este modelo, inspirado no modelo clássico de mesmo nome, considera que a fonte de tráfego alterna entre dois estados: o estado de atividade, no qual a fonte gera tráfego a uma taxa constante  $r$ , e o estado de silêncio, no qual nenhum tráfego é gerado. A duração destes estados é representada pelas variáveis aleatórias  $\tau_{on}$  e  $\tau_{off}$ , respectivamente. No modelo clássico, estas variáveis são independentes e podem assumir qualquer distribuição de decaimento exponencial; porém o modelo *on/off* de cauda longa considera que pelo menos uma destas variáveis tem distribuição de cauda longa, no sentido da Definição 2.9.

É interessante observar, contudo, que as estatísticas de primeira e segunda ordem do modelo *on/off* clássico e de cauda longa podem ser obtidas de modo semelhante, visto que não dependem das distribuições de  $\tau_{on}$  e de  $\tau_{off}$  propriamente ditas, mas apenas dos seus valores médios. De fato, o processo  $\mathbf{X}(t)$  que representa a quantidade de tráfego entre os instantes  $t - 1$  e  $t$  é um processo de Bernoulli, para o qual:

$$\begin{aligned}\mathbb{E}\{\mathbf{X}(t)\} &= rp \\ \text{Var}\{\mathbf{X}(t)\} &= r^2p(1 - p)\end{aligned}$$

onde  $p$  representa a probabilidade da fonte estar em atividade, i.e.

$$p = \frac{\mathbb{E}\{\tau_{on}\}}{\mathbb{E}\{\tau_{on}\} + \mathbb{E}\{\tau_{off}\}}$$

#### • Modelo M/Pareto/ $\infty$

O processo M/Pareto/ $\infty$  busca representar a auto-similaridade decorrente da agregação do tráfego de um número muito grande de fontes. Neste modelo, o tráfego de cada fonte chega em blocos ao servidor segundo um processo de Poisson de taxa  $\lambda$  e é atendido a uma taxa constante  $r$ . O tamanho dos blocos segue uma distribuição de Pareto, e não há restrição para o número de fontes que podem estar transmitindo simultaneamente — isto é justamente indicado pelo símbolo  $\infty$  na notação do processo.

Um processo M/Pareto/ $\infty$  discreto  $\mathbf{X}(t)$ , que representa a quantidade de tráfego entre os instantes  $t - 1$  e  $t$ , tem valor médio dado por:

$$\mathbb{E}\{\mathbf{X}(t)\} = \frac{\lambda r}{\gamma - 1}, \quad 1 < \gamma < 2$$

sendo  $\gamma$  o parâmetro da distribuição de Pareto, cuja relação com o parâmetro de auto-similaridade  $H$  é dada por  $\gamma = 3 - 2H$ .

#### • Movimento Browniano fracionário

O movimento Browniano fracionário foi inicialmente desenvolvido por Komolgorov, sendo formalmente definido por Mandelbrot e van Ness [24], que introduziram a sua atual denominação. A popularização deste

processo como modelo de tráfego auto-similar se deve, em grande parte, aos estudos de Norros [30]. O movimento Browniano fracionário será tratado com maior detalhe no Capítulo 3.

Embora os modelos de tráfego baseados em processos auto-similares sejam interessantes pela parcimônia na representação da natureza fractal, seu tratamento matemático não é trivial. Uma consequência natural disto é a busca de modelos intermediários, tais como os modelos pseudo auto-similares e os processos envelope. Neste estudo, será considerada a modelagem de tráfego pelo processo envelope do movimento Browniano fracionário [14, 26], a ser apresentado também no Capítulo 3.

## Capítulo 3

# O movimento Browniano fracionário

### 3.1 Introdução

O movimento Browniano fracionário (*fractional Brownian motion*, fBm) é um processo estocástico introduzido por Kolmogorov, e definido por Mandelbrot e van Ness [24] a partir do movimento Browniano puro:

**Definição 3.1.** Seja  $\mathbf{B}(t)$  um movimento Browniano puro. O movimento Browniano fracionário de expoente  $H \in (0; 1)$  é definido como o processo obtido a partir da média deslizante dos sucessivos incrementos  $d\mathbf{B}(t)$  ponderados pelo fator  $(t - s)^{H-1/2}$ , i.e.

$$\mathbf{Z}(t) = \frac{1}{\Gamma(H + \frac{1}{2})} \left\{ \int_{-\infty}^0 \left[ (t - s)^{H-1/2} - (-s)^{H-1/2} \right] d\mathbf{B}(s) + \int_0^t (t - s)^{H-1/2} d\mathbf{B}(s) \right\} \quad (3.1)$$

Pode-se verificar que o movimento Browniano fracionário se degenera no próprio movimento Browniano puro, se  $H = 1/2$ . A autocorrelação do processo  $\mathbf{Z}(t)$  é dada por:

$$\mathcal{R}_{\mathbf{Z}}(s, t) = \frac{1}{2} \left( |s|^{2H} + |t|^{2H} - |s - t|^{2H} \right) \quad (3.2)$$

ou seja, trata-se de um processo não-estacionário. Por suas características, o movimento Browniano fracionário é classificado como um processo de acumulação auto-similar, no sentido da Definição 2.1. Algumas de suas propriedades podem ser apontadas [30]:

1.  $\mathbf{Z}(t)$  é gaussiano e tem incrementos estacionários;
2. as realizações de  $\mathbf{Z}(t)$  são contínuas;

3.  $\mathbf{Z}(0) = 0$  e  $\mathbb{E}\{\mathbf{Z}(t)\} = 0$  para todo  $t$ ;
4.  $\mathbb{E}\{\mathbf{Z}^2(t)\} = |t|^{2H}$  para todo  $t$ .

O processo de incrementos  $\mathbf{X}(t)$ ,  $t \in \mathbb{Z}$  associado ao processo  $\mathbf{Z}(t)$  é denominado ruído Gaussiano fracionário (*fractional Gaussian noise*, fGn). Este processo, que é estacionário, tem média nula e função de autocorrelação dada por:

$$\mathcal{R}_{\mathbf{X}}(\tau) = \frac{1}{2} \left( |\tau + 1|^{2H} - 2|\tau|^{2H} + |\tau - 1|^{2H} \right)$$

É possível mostrar que, para cada valor de  $H \in (0; 1)$ , existe somente um único processo  $\mathbf{X}(t)$  gaussiano capaz de representar os incrementos do processo  $\mathbf{Z}(t)$ . Isto permite formular uma definição mais simples para o movimento Browniano fracionário, sem que seja necessário recorrer à integral (3.1) [36].

**Definição 3.2.** O processo  $\mathbf{Z}(t)$ ,  $t \in \mathbb{R}$  é denominado movimento Browniano fracionário com parâmetro  $H \in (0; 1)$  se  $\mathbf{Z}(t)$  é gaussiano e auto-similar.

**Definição 3.3.** O processo  $\mathbf{X}(t)$ ,  $t \in \mathbb{Z}_+$  é denominado ruído Gaussiano fracionário com parâmetro  $H \in (0; 1)$  se  $\mathbf{X}(t)$  equivale ao processo de incrementos associado a um processo do movimento Browniano fracionário.

*Comentário.* Dado que o processo  $\mathbf{X}(t)$  também é gaussiano, ele é completamente caracterizado pelas suas estatísticas de segunda ordem — desta forma, os conceitos de auto-similaridade (Definição 2.4) e auto-similaridade de segunda ordem (Definição 2.6) são equivalentes.

## 3.2 Síntese do ruído Gaussiano fracionário

Em geral, não é possível obter soluções analíticas exatas para diversos problemas envolvendo o movimento Browniano fracionário e o seu correspondente processo de incrementos, o ruído Gaussiano fracionário. Embora soluções de caráter assintótico possam ser obtidas utilizando a teoria dos valores extremos, muitos dos resultados ainda são obtidos por meio de simulação. Por este motivo, diversos trabalhos foram dedicados ao desenvolvimento de algoritmos para a síntese artificial daqueles processos.

A descrição dos algoritmos hoje disponíveis não estão no escopo deste trabalho. No entanto, podem ser citados os seguintes métodos de síntese de ruído Gaussiano fracionário:

- o algoritmo desenvolvido por Mandelbrot e posteriormente aperfeiçoado por Chi et al. [5];
- o algoritmo de Paxson, baseado na transformada rápida de Fourier [37, 38];

- o algoritmo de deslocamento do ponto médio aleatório (*random midpoint displacement*, RMD) desenvolvido por Lau et al. [21] e posteriormente aperfeiçoado por Norros et al. [31].

Outros métodos de interesse são descritos por Coeurjolly [6], que também trata da aproximação do movimento Browniano fracionário a partir das amostras do ruído gaussiano fracionário sintético.

### 3.3 O tráfego Browniano fracionário

Norros [30] propõe o seguinte processo para a modelagem de fontes de tráfego auto-similar:

**Definição 3.4.** Seja o processo

$$\mathbf{A}(t) = \rho t + \sigma \mathbf{Z}(t), \quad t \in \mathbb{Z}_+ \quad (3.3)$$

onde  $\mathbf{Z}(t)$  é um movimento Browniano fracionário de parâmetro  $H \in [1/2; 1)$ ,  $\rho > 0$  é a taxa média de chegada de tráfego e  $\sigma > 0$  é um coeficiente associado à variância incremental de  $\mathbf{A}(t)$ . Este processo de acumulação é denominado tráfego Browniano fracionário.

O processo  $\mathbf{A}(t)$  pode ser considerado um modelo parcimonioso, no qual o tráfego é caracterizado por apenas três parâmetros ( $\rho$ ,  $\sigma$  e  $H$ ). A sua aplicação na modelagem de tráfego pode ser justificada pela constatação de que a superposição de um grande número de fontes *on/off* identicamente distribuídas resulta em um movimento Browniano fracionário se, para cada fonte, a duração dos períodos de atividade e de silêncio assumir valores elevados com probabilidade não-negligenciável (efeito Noé) [44]. De fato, diversos estudos mostram que o tráfego Browniano fracionário é capaz de representar adequadamente o tráfego auto-similar, sobretudo quando ele apresenta características gaussianas. Tais características podem resultar tanto do comportamento de fontes individuais, como da agregação de um grande número de fontes independentes.

A capacidade de representação do tráfego Browniano fracionário é ilustrada, por exemplo, pela análise empírica de Norros [30]. Nesta análise, mostra-se que o tráfego Browniano fracionário é adequado à representação do tráfego na rede *ethernet* do *Bellcore Morristown Research and Engineering Center* (que servira de base para os estudos de Leland et al. [22]), cujas características gaussianas advém da agregação do tráfego de um grande número de fontes independentes. No entanto, a mesma análise mostra que este processo não é adequado à representação do tráfego no enlace de acesso externo do *Bellcore*, cuja reduzida agregação é insuficiente para compensar as suas características não-gaussianas.

### 3.3.1 Armazenamento do tráfego Browniano fracionário

Pode-se ainda definir um processo para descrever o tamanho da fila em um servidor que opera a uma taxa constante  $g$  e cujo tráfego acumulado é representado pelo processo  $\mathbf{A}(t)$  dado por (3.3):

**Definição 3.5.** Seja um servidor, de taxa constante  $g$ , alimentado por uma fonte cujo tráfego acumulado é representado pelo processo  $\mathbf{A}(t)$ , dado por (3.3). Se este servidor possui uma fila de capacidade infinita, o processo que caracteriza o seu comprimento no instante  $t$  é denominado processo de armazenamento do tráfego Browniano fracionário, e é dado por:

$$\mathbf{Q}(t) = \mathbf{A}(t) - gt, \quad t \in \mathbb{Z}_+ \quad (3.4)$$

A solução analítica para a distribuição de  $\mathbf{Q}(t)$  ainda não é conhecida, de modo que resultados são obtidos sobretudo por meio de simulação. No entanto, soluções de caráter assintótico podem ser obtidas analiticamente. Seja  $\mathbf{Q}^*$  a variável aleatória que representa o máximo comprimento da fila, i.e.

$$\mathbf{Q}^* = \max_{t \geq 0} \mathbf{Q}(t)$$

Desta forma, a distribuição de  $\mathbf{Q}^*$  é dada por:

$$\begin{aligned} \mathbb{P}\{\mathbf{Q}^* > x\} &= \mathbb{P}\left\{\max_{t \geq 0} \mathbf{Q}(t) > x\right\} \\ &= \mathbb{P}\left\{\max_{t \geq 0} [\mathbf{A}(t) - gt] > x\right\} \end{aligned} \quad (3.5)$$

Duffield e O'Connell obtiveram a solução para este problema de otimização utilizando a teoria dos valores extremos [11]. No entanto, Norros [29] propõe uma abordagem mais intuitiva. Seja o seguinte lema [11]:

**Lema 3.6.** *Seja  $\mathbf{V}(t)$  um processo estocástico discreto qualquer. Neste caso, pode-se dizer que:*

$$\mathbb{P}\left\{\max_{t \geq 0} \mathbf{V}(t) > x\right\} \geq \max_{t \geq 0} \mathbb{P}\{\mathbf{V}(t) > x\} \quad (3.6)$$

*Para valores grandes de  $x$ ,*

$$\mathbb{P}\left\{\max_{t \geq 0} \mathbf{V}(t) > x\right\} \simeq \max_{t \geq 0} \mathbb{P}\{\mathbf{V}(t) > x\} \quad (3.7)$$

O Lema 3.6 pode ser considerado uma consequência da estatística dos eventos raros: à medida em que  $x$  cresce, torna-se cada vez menos provável que o processo  $\mathbf{Q}(t)$  atinja esse valor; no entanto, se ele efetivamente o atinge, este evento ocorre no instante em que ele é mais provável. Adicionalmente, Duffield e O’Connell apresentam uma justificativa baseada no princípio de Laplace [11].

A relação (3.7) pode ser utilizada para caracterizar a distribuição de  $\mathbf{Q}^*$  para valores grandes de  $x$ . Neste caso, a relação (3.5) pode ser escrita da seguinte forma:

$$\mathbb{P}\{\mathbf{Q}^* > x\} \simeq \max_{t \geq 0} \mathbb{P}\{\mathbf{A}(t) - gt > x\}$$

que, por sua vez, equivale a:

$$\begin{aligned} \mathbb{P}\{\mathbf{Q}^* > x\} &\simeq \max_{t \geq 0} \mathbb{P}\{\rho t + \sigma \mathbf{Z}(t) - gt > x\} \\ &= \max_{t \geq 0} \mathbb{P}\{\rho t + \sigma t^H \mathbf{Z}(1) - gt > x\} \\ &= \max_{t \geq 0} \mathbb{P}\left\{\mathbf{Z}(1) > \frac{(g - \rho)t + x}{\sigma t^H}\right\} \\ &= \max_{t \geq 0} \overline{\mathbb{G}}\left(\frac{(g - \rho)t + x}{\sigma t^H}\right) \end{aligned} \quad (3.8)$$

lembrando que  $\mathbf{Z}(t) = t^H \mathbf{Z}(1)$  (Lema 2.2). A função  $\overline{\mathbb{G}}(y)$  representa a distribuição residual gaussiana, para a qual o seguinte lema pode ser estabelecido:

**Lema 3.7.** *Para valores grandes de  $x$ ,*

$$\overline{\mathbb{G}}(y) \sim \exp\left(-\frac{y^2}{2}\right) \quad (3.9)$$

Assim, segundo o Lema 3.7, a relação (3.8) pode ser aproximada por:

$$\mathbb{P}\{\mathbf{Q}^* > x\} \simeq \max_{t \geq 0} \exp\left\{-\frac{1}{2} \left[\frac{(g - \rho)t + x}{\sigma t^H}\right]^2\right\} \quad (3.10)$$

cujas solução é obtida em:

$$t = \frac{xH}{(g - \rho)(1 - H)} \quad (3.11)$$



O seguinte lema pode, então, ser estabelecido [29]:

**Lema 3.8.** *Seja um servidor, de taxa constante  $g$ , alimentado por uma fonte cujo tráfego acumulado é representado pelo processo  $\mathbf{A}(t)$ , dado por (3.3). Se este servidor possui uma fila de capacidade infinita, a distribuição da variável aleatória que representa o seu máximo comprimento é dada por:*

$$\mathbb{P}\{\mathbf{Q}^* > x\} \simeq \exp \left\{ -\frac{(g - \rho)^{2H}}{2\sigma^2 H^{2H} (1 - H)^{2-2H}} x^{2-2H} \right\} \quad (3.12)$$

*Comentário.* É interessante observar que (3.12) fornece a distribuição do comprimento da fila no instante em que ele atinge o seu valor máximo, mas não fornece nenhuma informação a respeito da distribuição de  $\mathbf{Q}(t)$  para um instante  $t$  arbitrário.

O Lema 3.8 indica que  $\mathbf{Q}^*$  segue uma distribuição Weibull, em particular para  $x \rightarrow \infty$ . A Figura 3.1 mostra esta distribuição para diversos valores de  $H$ . Pode-se verificar que, para processos com parâmetro  $H$  elevado, a distribuição residual de  $\mathbf{Q}^*$  decai muito lentamente, indicando que valores elevados são obtidos com probabilidade não-negligenciável.

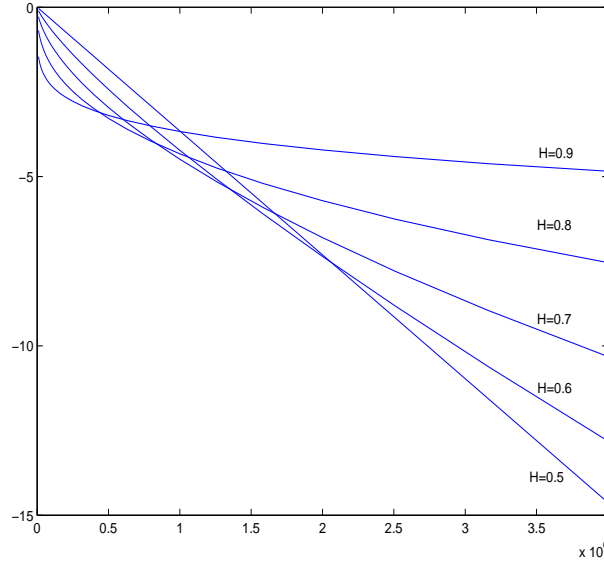


Figura 3.1: Aproximação Weibull de  $\log_{10} \mathbb{P}\{\mathbf{Q}^* > x\}$  para  $\rho = 2279kb/s$ ,  $\sigma = 262.8kb \cdot s$ ,  $g = 4.8Mb/s$  e diversos valores de  $H$  [30].

O resultado apontado pelo Lema 3.8 pode também ser utilizado para o dimensionamento da taxa do servidor em função da probabilidade de transbordo da fila, quando esta tem uma capacidade finita [30]. No contexto de redes, isto significa dimensionar a capacidade do enlace de um servidor em função do tamanho do seu *buffer* e da probabilidade de perda de pacotes desejada.

**Lema 3.9.** *Seja um servidor, de taxa constante  $g$ , alimentado por uma fonte cujo tráfego acumulado é representado pelo processo  $\mathbf{A}(t)$ , dado por (3.3). Se este servidor possui uma fila de capacidade  $x$  finita, não haverá transbordo com probabilidade  $1 - \varepsilon$ , i.e.  $\mathbb{P}\{\mathbf{Q}^* > x\} = \varepsilon$ , se:*

$$g = \rho + (-2\sigma^2 \ln \varepsilon)^{\frac{1}{2H}} H(1-H)^{\frac{1-H}{H}} x^{\frac{H-1}{H}}$$

### 3.3.2 Escala de Tempo Crítica

É interessante determinar, também, a escala de tempo na qual os transbordos são mais prováveis [27]. Seja  $\xi(t)$  o evento que representa, para cada instante  $t \in \mathbb{Z}_+$ , o transbordo da fila. Desta forma,

$$\begin{aligned} \max_{t \geq 0} \mathbb{P}\{\xi(t)\} &= \max_{t \geq 0} \mathbb{P}\{\mathbf{A}(t) - gt > x\} \\ &= \max_{t \geq 0} \mathbb{P}\{\rho t + \sigma \mathbf{Z}(t) > gt + x\} \\ &= \max_{t \geq 0} \mathbb{P}\left\{\mathbf{Z}(1) > \frac{t(g - \rho) + x}{\sigma}\right\} \end{aligned}$$

A escala de tempo na qual é mais provável a ocorrência de transbordos na fila, denominada Escala de Tempo Crítica (*Critical Time Scale*, CTS), pode ser obtida resolvendo o seguinte problema de otimização:

$$\begin{aligned} t_{cts} &= \arg \max_{t \geq 0} \mathbb{P}\{\xi(t)\} \\ &= \arg \max_{t \geq 0} \mathbb{P}\left\{\mathbf{Z}(1) > \frac{t(g - \rho) + x}{\sigma}\right\} \\ &= \arg \max_{t \geq 0} \overline{\mathbb{G}}\left[\frac{t(g - \rho) + x}{\sigma}\right] \end{aligned}$$

Para valores grandes de  $x$ , e assumindo que  $g > \rho$  (caso contrário o sistema é instável), pode-se recorrer ao Lema 3.7 para aproximar a distribuição residual Gaussiana, de modo que:

$$t_{cts} \simeq \arg \max_{t \geq 0} \exp\left\{-\frac{1}{2} \left[\frac{t(g - \rho) + x}{\sigma}\right]^2\right\}$$

**Definição 3.10.** (Escala de Tempo Crítica) Seja um sistema fila-servidor que opera sob taxa constante  $g$  e cuja fila tenha capacidade finita, dada por  $x$ . Se este sistema é alimentado por uma fonte cujo tráfego é representado

pelo processo  $\mathbf{A}(t)$ , dado por (3.3), a escala de tempo na qual é mais provável a ocorrência de transbordos é denominada Escala de Tempo Crítica, que é definida como:

$$t_{cts} = \frac{xH}{(g - \rho)(1 - H)}$$

Segundo Ryu et al. [40],  $t_{cts}$  representa a escala de tempo mais importante na determinação de perdas de células em redes ATM. É importante observar que esta escala de tempo está associada somente com a ocorrência de transbordo da fila no instante em que ele é mais provável e não permite, portanto, caracterizar a probabilidade de transbordo em outros instantes de tempo [25].

### 3.4 O processo envelope do movimento Browniano fracionário

Em geral, o uso do tráfego Browniano fracionário em problemas de análise de desempenho de redes não é trivial, devido ao seu difícil tratamento matemático. De fato, o uso de outros modelos baseados em processos estocásticos apresenta dificuldades semelhantes, justificando-se a busca por outras formas de representar o tráfego nas redes. Uma possibilidade é o uso de modelos cujo objetivo não é caracterizar de modo exato o tráfego, mas representá-lo por meio de uma função limitante. Esta função, geralmente de tratamento matemático mais simples, é denominada Processo Envelope.

Diversos modelos de tráfego baseados em processos envelope foram recentemente propostos; alguns deles são apresentados em [43]. É possível definir um processo envelope para o tráfego Browniano fracionário que mantenha a maioria das suas propriedades, mas com uma complexidade matemática menor [14, 26, 43]:

**Definição 3.11.** (Processo envelope do movimento Browniano fracionário) Seja o tráfego Browniano fracionário  $\mathbf{A}(t)$ , dado por (3.3). A este processo pode ser associado o processo envelope

$$\hat{A}(t) = \rho t + k\sigma t^H, \tag{3.13}$$

onde o parâmetro  $k$  determina a probabilidade do processo  $\mathbf{A}(t)$  violar o envelope  $\hat{A}(t)$  no instante  $t$ , i.e.

$$\begin{aligned} \mathbb{P}\{\mathbf{A}(t) > \hat{A}(t)\} &= \mathbb{P}\{\rho t + \sigma \mathbf{Z}(t) > \rho t + k\sigma t^H\} \\ &= \mathbb{P}\{\mathbf{Z}(1) > k\} \\ &= \overline{\mathbb{G}}(k) \end{aligned} \tag{3.14}$$

Segundo o Lema 3.7:

$$\mathbb{P}\left\{\mathbf{A}(t) > \widehat{A}(t)\right\} \simeq \exp\left(-\frac{k^2}{2}\right), \quad (3.15)$$

para valores suficientemente grandes de  $k$ .

Algumas peculiaridades do processo envelope do movimento Browniano fracionário são apresentadas em [14, 43], onde a sua eficácia na representação de processos fBm sintéticos e de seqüências MPEG foi extensivamente comprovada.

Dentre suas vantagens, pode-se destacar que o processo envelope mantém o caráter parcimonioso do tráfego Browniano fracionário, pois apenas três parâmetros são necessários para caracterizar completamente o tráfego<sup>1</sup>. A sua complexidade matemática é, entretanto, muito menor. Além disso, ele pode ser utilizado para representar tráfego com dependência tanto de longa como de curta duração, sendo necessário apenas ajustar os parâmetros de maneira adequada. Neste sentido, um método simples para determinar os parâmetros do processo envelope para uma seqüência  $A(t)$ , correspondente à realização de um processo  $\mathbf{A}(t)$  qualquer, é proposto por Fonseca et al. [14].

### 3.4.1 Armazenamento do processo envelope do fBm

Como no caso do tráfego fBm original, é possível definir um processo envelope para descrever o tamanho da fila em um servidor que opera a uma taxa constante  $g$  e que é alimentado por tráfego fBm:

**Definição 3.12.** Seja um servidor, de taxa constante  $g$ , alimentado por uma fonte cujo tráfego acumulado é representado pelo processo envelope  $\widehat{A}(t)$ , dado por (3.13). Se este servidor possui uma fila de capacidade infinita, o processo envelope que caracteriza o seu comprimento no instante  $t$  é dado por:

$$\widehat{Q}(t) = (\rho - g)t + k\sigma t^H, \quad t \in \mathbb{Z}_+ \quad (3.16)$$

Se o processo que alimenta a fila for tráfego Browniano fracionário, então o seu comprimento pode ser representado pelo processo  $\mathbf{Q}(t)$ , dado por (3.4). Neste caso, é fácil constatar que a probabilidade de que, no instante  $t$ , o tamanho da fila ultrapasse  $\widehat{Q}(t)$  é dada por:

$$\begin{aligned} \mathbb{P}\left\{\mathbf{Q}(t) > \widehat{Q}(t)\right\} &= \mathbb{P}\left\{\mathbf{A}(t) - gt > \widehat{A}(t) - gt\right\} \\ &= \mathbb{P}\left\{\mathbf{A}(t) > \widehat{A}(t)\right\} \\ &= \overline{\mathbb{G}}(k) \end{aligned} \quad (3.17)$$

---

<sup>1</sup>Considera-se o produto  $k\sigma$  como um único parâmetro.

Para valores grandes de  $k$ ,  $\mathbb{P}\{\mathbf{Q}(t) > \widehat{Q}(t)\} \simeq \exp(-k^2/2)$ . (Lema 3.7).

### 3.4.2 Escala de tempo máxima

Mayor and Silvester [26] definiram uma escala de tempo de interesse para sistemas de filas alimentados por tráfego descrito pelo processo envelope do movimento Browniano fracionário, denominada Escala de Tempo Máxima (*Maximum Timescale*, MaxTs). Esta escala de tempo representa o instante de tempo no qual o volume de tráfego em espera em um sistema de filas atinge, no sentido probabilístico, o seu valor máximo. Após este instante, a taxa média de chegadas cai abaixo da taxa de serviço, e o tamanho da fila começa a diminuir.

Matematicamente, a escala de tempo MaxTS é definida como:

**Definição 3.13.** (Escala de Tempo Máxima) Seja um sistema fila-servidor que opera sob taxa constante  $g$  e cuja fila tenha capacidade infinita. Se este sistema é alimentado por uma fonte cujo tráfego acumulado é representado pelo processo envelope  $\widehat{A}(t)$ , dado por (3.13), então a escala de tempo na qual o comprimento da fila atinge, no sentido probabilístico, o seu valor máximo é denominada Escala de Tempo Máxima e é definida como:

$$t^* = \arg \max_{t \geq 0} \widehat{Q}(t) = \left[ \frac{k\sigma H}{g - \rho} \right]^{\frac{1}{1-H}}$$

**Lema 3.14.** *Seja um sistema fila-servidor que opera sob taxa constante  $g$  e cuja fila tenha capacidade infinita. Se este sistema é alimentado por uma fonte cujo tráfego acumulado é representado pelo processo envelope  $\widehat{A}(t)$ , dado por (3.13), então o comprimento máximo da fila, no sentido probabilístico, é dado por:*

$$Q^* = \max_{t \geq 0} \widehat{Q}(t) = (g - \rho)^{\frac{H}{H-1}} (k\sigma)^{\frac{1}{1-H}} H^{\frac{H}{1-H}} (1 - H) \quad (3.18)$$

sendo que:

$$\begin{aligned} \mathbb{P}\{\mathbf{Q}(t) > Q^*\} &\leq \mathbb{P}\{\mathbf{Q}(t) > \widehat{Q}(t)\} \\ &= \overline{\mathbb{G}}(k) \end{aligned} \quad (3.19)$$

se o processo que alimenta a fila for tráfego Browniano fracionário, de modo o comprimento da fila possa ser representado pelo processo  $\mathbf{Q}(t)$ , dado por (3.4).

A relação (3.19) indica que, com probabilidade maior ou igual a  $1 - \overline{\mathbb{G}}(k)$ , o comprimento da fila não será maior que  $Q^*$ . Cabe lembrar que, para valores elevados de  $k$ , (3.19) pode ser aproximada por  $\exp(-k^2/2)$  (Lema 3.7).

As relações (3.18) e (3.19) são de especial interesse no contexto da análise de redes, pois juntas constituem um limitante para o volume de tráfego em espera no *buffer* de um servidor que opera a taxa constante, cujo tratamento matemático é bem mais simples que aquele dispensado à análise por meio da modelagem utilizando o tráfego Browniano fracionário original.

O resultado apontado pelo Lema 3.14 pode também ser utilizado para o dimensionamento da taxa do servidor em função da probabilidade de transbordo da fila, quando esta tem uma capacidade finita  $x$ . No contexto de redes, isto significa dimensionar a capacidade do enlace de um servidor em função do tamanho do seu *buffer* e da probabilidade de perda de pacotes desejada [43].

**Lema 3.15.** *Seja um servidor, de taxa constante  $g$ , alimentado por uma fonte cujo tráfego acumulado é representado pelo processo envelope fBm que, segundo a Definição 3.11, é dado por (3.13). Se este servidor possuir uma fila de capacidade  $x$  finita, não haverá transbordo com probabilidade maior ou igual a  $1 - \varepsilon$ , i.e.  $\mathbb{P}\{\mathbf{Q}(t) > x\} \leq \varepsilon$  se:*

$$g = \rho + (k\sigma)^{\frac{1}{2H}} H(1 - H)^{\frac{1-H}{H}} x^{\frac{H-1}{H}}$$

sendo  $\varepsilon = \overline{\mathbb{G}}(k)$  ou, para valores grandes de  $k$ ,  $\varepsilon \simeq \exp(-k^2/2)$  (Lema 3.7).

*Comentário.* Este resultado é semelhante àquele obtido por Norros [30] utilizando o tráfego Browniano fracionário (Lema 3.9). No entanto, ele é obtido de maneira mais simples e não é baseado em aproximações assintóticas. Isto ilustra como o processo envelope do movimento Browniano fracionário é capaz de manter as mesmas propriedades do tráfego Browniano fracionário, mas com um tratamento matemático mais simples.

Finalmente, pode-se observar que o raciocínio utilizado para definir o conceito de Escala de Tempo Máxima pode ser aplicado também para solucionar problemas de multiplexação de tráfego auto-similar em sistemas fila-servidor. Neste caso, pode-se analisar o comportamento da fila quando o sistema é alimentado por múltiplas fontes heterogêneas, estabelecendo relações limitantes que podem ser utilizadas no desenvolvimento de mecanismos de controle de admissão. Neste sentido, destacam-se os estudos de Fonseca et al. [13, 14].

## Capítulo 4

# Policiamento de tráfego auto-similar

O controle de admissão não é suficiente para garantir a qualidade do serviço prestado pela rede, visto que os usuários podem descumprir o contrato estabelecido na ocasião de sua admissão. É necessário, portanto, que o tráfego dos usuários seja constantemente monitorado, e que as violações de contrato sejam tratadas de forma adequada. Este procedimento é denominado policiamento de tráfego ou controle de parâmetros de usuário (*user parameters control*, UPC).

Para que seja considerado eficiente, o policiamento de tráfego deve ser transparente aos usuários cujo tráfego esteja em acordo com os parâmetros negociados, mas deve também ser capaz de detectar e tomar ações corretivas na ocorrência de violações de contrato. Neste sentido, são basicamente três as ações que podem ser tomadas pelo mecanismo de policiamento:

- descartar a parcela excedente do tráfego, correspondente à violação do contrato do usuário;
- marcar a parcela excedente do tráfego, de forma a priorizá-la para descarte em caso de congestionamento;
- atrasar a parcela excedente do tráfego, de forma a forçar o cumprimento do contrato.

Mecanismos de policiamento que adotam esta última ação são denominados reguladores ou conformadores de tráfego. Existem atualmente diversas técnicas para realizar o policiamento, dentre as quais se destacam as técnicas baseadas em janelas e o algoritmo do Balde Furado (*Leaky Bucket*). Em geral, o algoritmo do Balde Furado apresenta desempenho superior às técnicas baseadas em janelas [18], sendo recomendado pelo ITU-T e ATM Forum como técnica de policiamento para redes ATM, na forma do *Generic Cell Rate Algorithm (GCR)* [41].

Sob tráfego auto-similar, entretanto, o algoritmo do Balde Furado não pode ser considerado eficaz [13]. Diante deste fato, e da constatação de que, no futuro, a maior parte do tráfego das redes será de natureza auto-similar, torna-se necessária a elaboração de novos mecanismos de policiamento. Neste sentido, Fonseca et al. [13, 14] introduzem um mecanismo de policiamento baseado na caracterização do tráfego pelo processo envelope do

movimento Browniano fracionário. Este mecanismo, denominado algoritmo do Balde Furado Fractal, é discutido na Seção 4.2.2.

## 4.1 Técnicas de policiamento de tráfego

### 4.1.1 Técnicas de policiamento baseadas em janelas

Mecanismos de policiamento podem ser implementados utilizando diversas técnicas baseadas em janelas. Dentre elas, a mais simples é a técnica de janela saltitante (*jumping window*), na qual são definidas janelas de tempo consecutivas e não-sobrepostas de tamanho fixo, sendo imposto um volume máximo de tráfego que o mecanismo de policiamento deve aceitar dentro de cada janela. O volume excedente é descartado ou marcado como prioritário para descarte.

Seja  $m$  o volume máximo de tráfego que pode ser aceito em cada janela, cujo tamanho é dado por  $T$ . O mecanismo de policiamento de janelas saltitantes atua no sentido de controlar a taxa de pico ou a taxa média do tráfego (não simultaneamente) de forma que ela seja menor que  $m/T$ .

A probabilidade de que o mecanismo de policiamento descarte tráfego de um usuário que cumpre o contrato é dada por [18]:

$$p = \frac{\int_{i=0}^{\infty} i x_{m+i}(T) di}{\int_{i=0}^{\infty} i x_i(T) di}$$

onde  $x_i(T)$  é a probabilidade de que o volume de tráfego gerado no intervalo de tempo  $T$  seja igual a  $i$ . Esta relação pode ser utilizada para encontrar o tamanho ótimo da janela, de forma que o valor de  $p$  seja muito pequeno. A partir deste valor de  $T$ , o parâmetro  $m$  é dado por  $m = T\rho_{js}$ , sendo  $\rho_{js}$  o valor desejado para a taxa a ser controlada.

Cabe observar, entretanto, que esta técnica de policiamento não é recomendada quando a transparência aos usuários em conformidade só pode ser atingida utilizando janelas muito grandes pois, neste caso, o tempo necessário para a detecção de violações torna-se alto e proibitivo.

A técnica de janelas saltitantes apresenta algumas variações que podem ser destacadas:

- janela saltitante com gatilho (*triggered jumping window*). As janelas são sincronizadas com a atividade da fonte, de modo que uma nova janela só se inicia com a chegada de tráfego. Assim, janelas consecutivas não são necessariamente consecutivas no tempo;
- janela deslizante (*moving window*). Similar à técnica de janela saltitante, mas considera que a janela se move de modo contínuo no tempo;



- média móvel ponderada exponencialmente (*exponentially weighted moving average*). Neste mecanismo, o volume de tráfego aceito durante uma janela é dado por uma soma ponderada do volume aceito em janelas anteriores.

#### 4.1.2 Algoritmo do Balde Furado

O algoritmo do Balde Furado é o mecanismo de policiamento mais conhecido e estudado na literatura. Ele é baseado no conceito de permissões (*tokens*), de modo que um pacote é aceito na rede somente se ele puder obter um número suficiente de permissões do mecanismo de policiamento. Estas permissões são geradas a uma taxa  $\bar{\rho}$  constante, denominada taxa de drenagem (*link rate*), e são armazenadas dentro de um reservatório (*token pool*) de tamanho  $\bar{\sigma}$  limitado. Permissões geradas quando o reservatório está cheio são descartadas.

Do ponto de vista estrutural, o algoritmo do Balde Furado é apresentado na Figura 4.1, onde são mostradas as possibilidades de tratamento da parcela excedente de tráfego, correspondente à violação de contrato.

O algoritmo do Balde Furado também pode ser implementado por um contador que é incrementado a cada chegada de pacote, e decrementado periodicamente com taxa  $\bar{\rho}$ . Neste caso, as ações de policiamento se aplicam aos pacotes que chegam quando o contador atinge um valor máximo  $\bar{\sigma}$ .

Como mecanismo de policiamento, o algoritmo do Balde Furado pode ser utilizado tanto para controlar a taxa de pico de uma fonte como a taxa média, sendo os parâmetros  $\bar{\rho}$  e  $\bar{\sigma}$  escolhidos de acordo com o objetivo desejado. Viana Neto [43] apresenta um interessante resumo de diversos estudos relacionados a este algoritmo.

#### 4.1.3 Análise do algoritmo do Balde Furado

O algoritmo do Balde Furado pode ser modelado como uma fila  $G/D/1/N$  [18]; nesta seção, entretanto, propõe-se uma outra forma de análise, mais adequada ao contexto deste trabalho. Para uma sessão  $i$  qualquer,  $A_i(\tau; t)$  representa o volume de tráfego que sai do mecanismo de policiamento no intervalo  $[\tau; t]$ . A variável  $\tau$  é omitida sempre que seu valor for zero. De modo a simplificar a análise, assume-se que:

**Condição 4.1.** O mecanismo de policiamento sempre descarta o volume de tráfego correspondente às violações de contrato;

**Condição 4.2.** O atraso de propagação nos enlaces é desprezível e, portanto, não influencia a operação do mecanismo de policiamento.

Sob estas condições, o mecanismo do Balde Furado tende a impor ao tráfego da sessão  $i$  a seguinte restrição [20]:

$$A_i(\tau; t) \leq \bar{\rho}_i (t - \tau) + \bar{\sigma}_i, \forall t \geq \tau \geq 0 \quad (4.1)$$

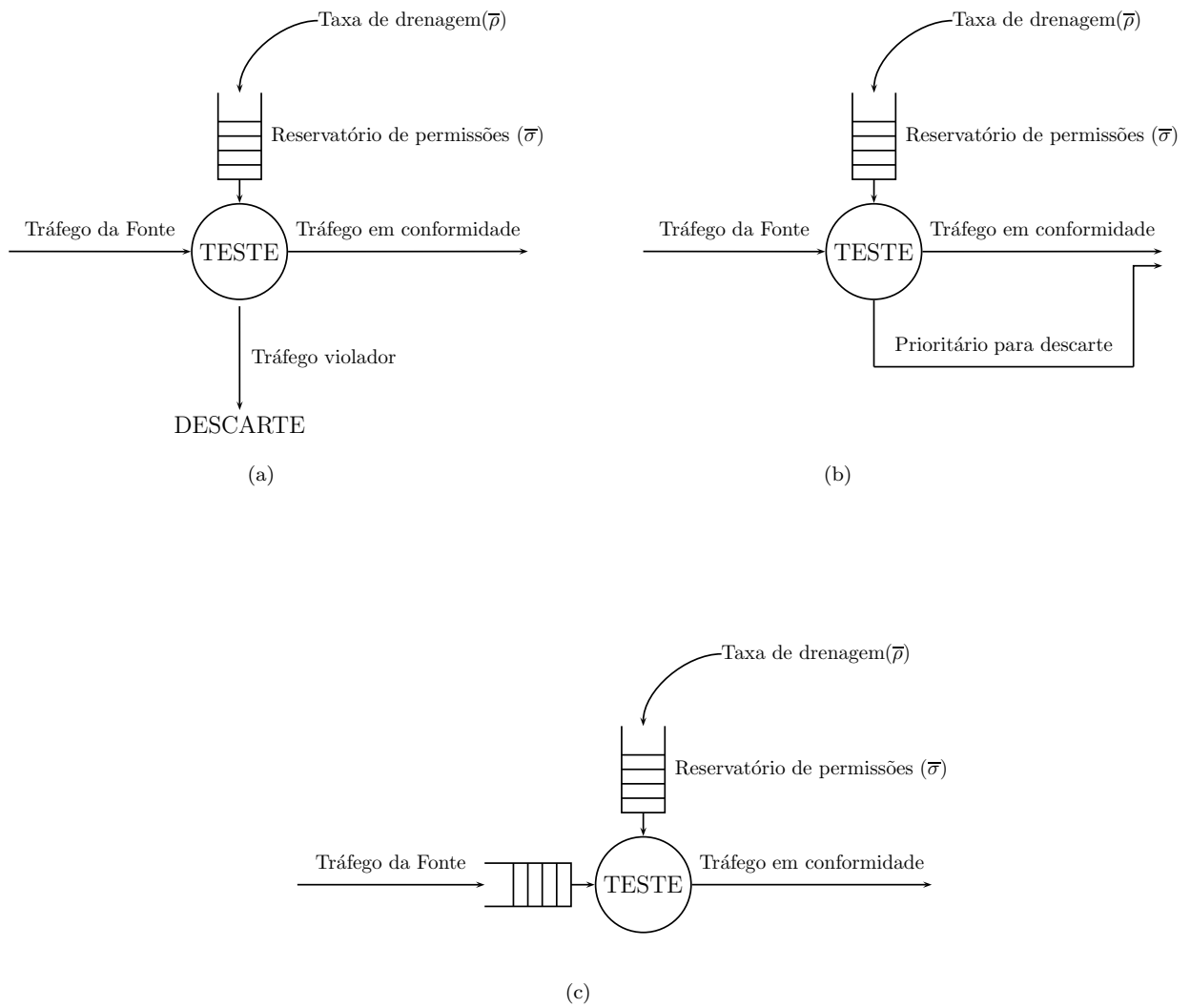


Figura 4.1: O algoritmo do Balde Furado e as possibilidades de tratamento da parcela de tráfego correspondente a violações: (a) descarte; (b) marcação para que seja prioritária no descarte; e (c) armazenamento para posterior transmissão.

sendo  $\bar{\rho}_i$  e  $\bar{\sigma}_i$  respectivamente a taxa de drenagem e o tamanho do reservatório de permissões do mecanismo de policiamento da sessão  $i$ . Desta forma, o tráfego que deixa o mecanismo de policiamento pode ser representado por um processo de chegadas com limitante linear (*linear bounded arrival process*, LBAP), o que é representado matematicamente como  $A_i \sim (\bar{\rho}_i; \bar{\sigma}_i)$  [9, 10].

A Figura 4.2 mostra a restrição imposta pelo algoritmo do Balde Furado ao tráfego, sendo  $s_i(t)$  o volume de permissões disponíveis no reservatório da sessão  $i$  no instante  $t$ , e  $l_i(t)$  o volume de permissões adicionado ao reservatório no intervalo  $(0; t]$ .

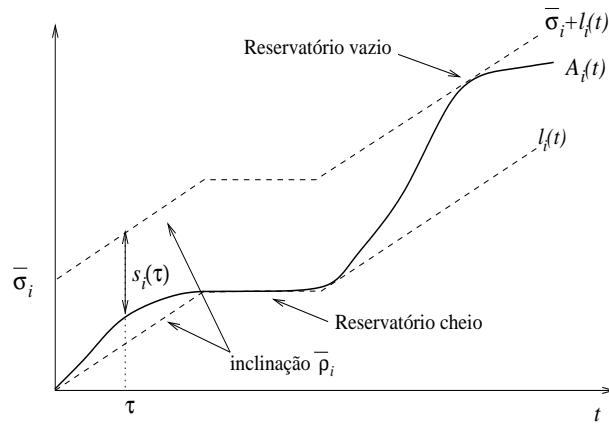


Figura 4.2: Restrição imposta ao tráfego pelo algoritmo do Balde Furado [32].

A capacidade limitada do reservatório de permissões confere ao algoritmo do Balde Furado uma importante propriedade. Suponha que, no instante inicial  $\tau$  o reservatório de permissões encontra-se completamente cheio, de modo que a expressão no lado direito da relação (4.1) represente o total de permissões disponibilizadas pelo mecanismo de policiamento para o intervalo  $[\tau; t]$ . Sob esta condição, pode-se dizer que a igualdade em (4.1) deixa de ser realizável caso ocorra o descarte de permissões pelo transbordo do reservatório. Isto pode acontecer, por exemplo, se a sessão  $i$  gera tráfego de forma que:

$$A_i(\tau; t') < \bar{\rho}_i(t' - \tau) + \bar{\sigma}_i, \text{ para algum } t' \in (\tau; t]$$

Isto significa que uma sessão não pode operar sob condição de estrita desigualdade em (4.1) durante um intervalo de tempo  $(\tau; t']$ , sendo  $t' \in (\tau; t]$  e, nos instantes seguintes, produzir excesso de tráfego de modo a atingir a igualdade daquela relação no instante  $t$ . Tal condição só pode ser atingida se:

$$A_i(\tau; t') \geq \bar{\rho}_i(t' - \tau), \forall t' \in (\tau; t]$$

Desta forma, pode-se concluir que o máximo volume de tráfego em excesso à média que é aceito pelo algoritmo do Balde Furado durante o intervalo  $(\tau; t']$  nas condições estabelecidas é limitado por  $\bar{\sigma}_i$ . Este raciocínio pode ser facilmente estendido para o caso do reservatório não estar completamente cheio no instante  $\tau$ , mas contiver um volume  $s_i(\tau)$  de permissões. Neste instante, o maior surto que pode ser aceito é dado justamente por  $s_i(\tau)$  que, por sua vez, é menor que  $\bar{\sigma}_i$ . Isto significa que o algoritmo do Balde Furado não é só capaz de restringir o tráfego da sessão  $i$  ao envelope linear dado pela expressão no lado direito da relação (4.1), mas também de limitar a sua explosividade, de forma que:

$$A_i(\tau; t) \leq \bar{\rho}_i(t - \tau) + s_i(\tau), \forall t \geq \tau \geq 0 \quad (4.2)$$

A relação (4.2) permite obter o envelope que representa o máximo volume de tráfego aceito pelo algoritmo do Balde Furado. Sejam as seguintes definições:

**Definição 4.3.** (*Sessões exatamente conformes*) Uma sessão  $i$  qualquer é considerada exatamente conforme (ou *greedy*, segundo Parekh e Gallager [32]) desde o instante  $\tau$  se o seu tráfego acumulado é dado, a partir deste instante, exatamente pelo envelope do mecanismo de policiamento.

**Definição 4.4.** (*Envelope do algoritmo do Balde Furado*). Seja o seguinte processo envelope:

$$\hat{A}_{(bf)}(t) = \bar{\rho}t + \bar{\sigma}, \forall t \geq 0 \quad (4.3)$$

Este processo é denominado processo envelope do algoritmo do Balde Furado, sendo os parâmetros  $\bar{\rho}$  e  $\bar{\sigma}$  correspondentes, respectivamente, à taxa de drenagem e ao tamanho do reservatório de permissões utilizados no algoritmo.

Para o caso do algoritmo do Balde Furado, a Definição 4.3 é ilustrada pela Figura 4.3: até o instante  $\tau$ , o tráfego gerado pela sessão  $i$  é tal que o reservatório de permissões nunca se encontra vazio. No instante  $\tau$ , entretanto, esta sessão aumenta subitamente o seu tráfego de modo a consumir todas as permissões disponíveis, e o tráfego da sessão  $i$  passa a ser limitado pela taxa de drenagem  $\bar{\rho}_i$ . Obviamente, o consumo instantâneo de todas as permissões do reservatório no instante  $\tau$  só é possível sob a Condição 4.2. No entanto, é possível mostrar que ela representa a situação de “pior caso” para a análise a seguir [32].

Seja  $A_i^-(t)$  uma possível realização do processo  $\mathbf{A}_i(t)$  para a qual o tráfego da sessão  $i$  é exatamente conforme ao envelope do algoritmo do Balde Furado desde o instante  $\tau$ . A partir de (4.2), pode-se verificar que:

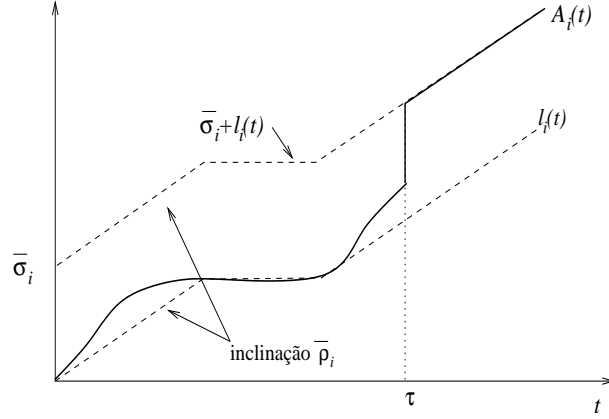


Figura 4.3: Sessão exatamente conforme ao envelope do algoritmo do Balde Furado a partir do instante  $\tau$  [32].

$$A_i^0(t) \geq A_i^\tau(t), \forall A_i^\tau \sim (\bar{\rho}_i; \bar{\sigma}_i)$$

ou seja, o envelope que representa o máximo volume de tráfego que deixa o mecanismo de policiamento é obtido quando a sessão  $i$  é exatamente conforme ao envelope do algoritmo do Balde Furado a partir do instante zero. Este resultado, em conjunto com (4.3), permite estabelecer o seguinte lema:

**Lema 4.5.** *Após o policiamento pelo algoritmo do Balde Furado, o tráfego da sessão  $i$  pode ser representado pelo envelope apresentado na Definição 4.4, i.e.,*

$$A_i(t) \leq \hat{A}_{i(bf)}(t), \forall t \geq 0 \quad (4.4)$$

i.e.,  $A_i(t) \sim \hat{A}_{i(bf)}(t)$ . A condição de igualdade em (4.4) somente é atingida se a sessão  $i$  é exatamente conforme ao envelope desde o instante zero.

## 4.2 Policiamento de tráfego auto-similar

Nesta seção, será analisado o problema do policiamento de uma sessão  $i$  cujo tráfego apresenta auto-similaridade. Assume-se que o seu tráfego é representado pelo processo envelope do movimento Browniano fracionário  $\hat{A}_i(t)$ , dado por (3.13), e cujos parâmetros são negociados com a rede na fase de admissão e devem ser utilizados para determinar os parâmetros do mecanismo de policiamento.

Supõe-se ainda que:

**Condição 4.6.** O tráfego da sessão  $i$  é exatamente conforme ao envelope do mecanismo de policiamento desde o instante zero.

#### 4.2.1 Policiamento utilizando o algoritmo do balde furado

Diversos estudos mostram que pode ser muito difícil realizar o policiamento de fontes cujo tráfego tem muitos surtos utilizando o algoritmo do Balde Furado [7]. Em particular, o desempenho deste algoritmo sob tráfego auto-similar foi analisado por Viana Neto [43], considerando a representação do tráfego da sessão  $i$  por meio do processo envelope do movimento Browniano fracionário (Definição 3.11). Na sua análise, Viana Neto mostra que os parâmetros do algoritmo de policiamento podem ser obtidos a partir dos parâmetros do processo  $\hat{A}_i(t)$ , dado por (3.13), por meio da seguinte relação:

$$(\rho_i - \bar{\rho}_i) \left( \frac{k_i \sigma_i H_i}{\bar{\rho}_i - \rho_i} \right)^{\frac{1}{1-H_i}} + k_i \sigma_i \left( \frac{k_i \sigma_i H_i}{\bar{\rho}_i - \rho_i} \right)^{\frac{H_i}{1-H_i}} - \bar{\sigma}_i \leq 0 \quad (4.5)$$

que pode ser utilizada para determinar o tamanho do reservatório de permissões ( $\bar{\sigma}_i$ ) para uma dada taxa de drenagem ( $\bar{\rho}_i$ ), ou vice-versa. Resultados numéricos mostram que valores proibitivos para  $\bar{\sigma}_i$  são obtidos se o tráfego apresentar grande auto-similaridade (i.e. valores elevados para o parâmetro  $H_i$ ) e o valor escolhido para  $\bar{\rho}_i$  for muito próximo da taxa média do tráfego. Obviamente, a escolha de valores mais elevados para  $\bar{\rho}_i$  pode reduzir significativamente o tamanho do reservatório, mas isto significa supor que o tráfego tem uma taxa média maior do que a declarada, o que reduz a eficiência da multiplexação estatística nos comutadores.

Apesar dos resultados importantes, a análise de Viana Neto é focalizada no problema de garantir a transparência do mecanismo de policiamento ao tráfego que não constitua violação — de fato, a relação (4.5), não permite caracterizar o seu comportamento frente a violações. No entanto, o processo envelope  $\hat{A}_i(t)$  é uma função estritamente convexa para  $t > 0$ , e uma reta que lhe seja tangente em um ponto  $t = t^*$  qualquer é sempre maior ou igual a esta função. Isto permite formular a seguinte proposição:

**Proposição 4.7.** *Para uma sessão  $i$  qualquer, os parâmetros do algoritmo do Balde Furado devem ser ajustados de modo que o envelope deste algoritmo seja equivalente a uma reta tangente, em um ponto  $t^*$  qualquer, ao processo envelope  $\hat{A}_i(t)$ , dado por (3.13) .*

Esta proposição é ilustrada na Figura 4.4. É possível mostrar que a reta tangente à função  $\hat{A}_i(t)$  em um ponto  $t = t^*$  qualquer, sendo  $t^* > 0$ , é dada por:

$$r_i(t) = (\rho_i + k_i \sigma_i H_i t^{*H_i-1}) t + (1 - H_i) k_i \sigma_i t^{*H_i}$$

Fazendo de  $r_i(t)$  o envelope do algoritmo do Balde Furado, i.e.,  $\hat{A}_{i(bf)}(t) = r_i(t)$ , pode-se estabelecer o seguinte lema:

**Lema 4.8.** *Os parâmetros do algoritmo do Balde Furado para uma sessão  $i$  qualquer, cujo tráfego é representado pelo processo envelope do movimento Browniano fracionário  $\hat{A}_i(t)$ , são dados por:*

$$\begin{aligned}\bar{\rho}_i &= \rho_i + k_i \sigma_i H_i t^{*H_i-1} \\ \bar{\sigma}_i &= (1 - H_i) k_i \sigma_i t^{*H_i}\end{aligned}$$

para um valor de  $t^* > 0$  qualquer.

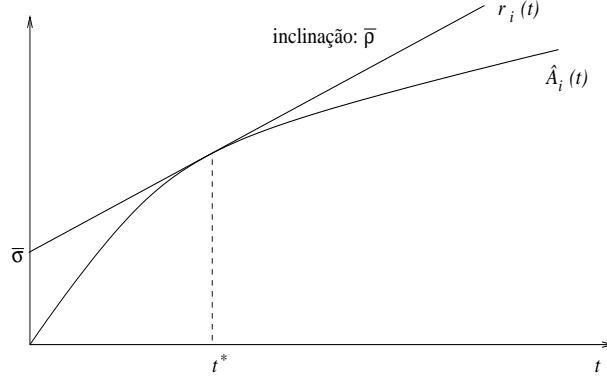


Figura 4.4: Uso de retas tangentes ao processo  $\hat{A}_i(t)$  no ajuste do algoritmo do Balde Furado.

O Lema 4.8 indica que os parâmetros do algoritmo do Balde Furado poderiam ser obtidos por meio da escolha de um valor adequado de  $t^*$ , visando a uma solução de compromisso entre o tamanho do reservatório (crescente com  $t^*$ ) e a taxa de drenagem (decrescente com  $t^*$ ). É interessante observar que os parâmetros obtidos a partir deste método correspondem à condição de igualdade na relação (4.5) proposta por Viana Neto [43], mas têm a vantagem de dispensar, em princípio, o uso de métodos numéricos no cálculo de  $\bar{\rho}_i$  e  $\bar{\sigma}_i$ .

O ajuste dos parâmetros pela reta tangente permite identificar as razões da ineficácia do algoritmo do Balde Furado no policiamento de tráfego auto-similar: sendo os parâmetros do algoritmo ajustados de acordo com o Lema 4.8, tem-se que  $\hat{A}_{i(bf)}(t) \geq \hat{A}_i(t)$ ,  $\forall t \geq 0$ , sendo a igualdade atingida somente no ponto de tangência. Para os demais valores de  $t$ ,  $\hat{A}_{i(bf)}(t)$  é estritamente maior que  $\hat{A}_i(t)$ . Isto significa que o tráfego da sessão pode violar o envelope  $\hat{A}_i(t)$  e, ainda assim, ser aceito pelo algoritmo do Balde Furado. Obviamente, empregar valores de  $\bar{\rho}_i$  e  $\bar{\sigma}_i$  menores que aqueles estabelecidos no Lema 4.8 não permite garantir a transparência do mecanismo de policiamento aos usuários em conformidade com o contrato.

### 4.2.2 Algoritmo do Balde Furado Fractal

O algoritmo do Balde Furado Fractal foi a proposta elaborada para suplantiar a ineficácia do algoritmo do Balde Furado no policiamento de tráfego auto-similar [14, 26]. Este algoritmo restringe o tráfego de uma sessão  $i$  ao processo envelope do movimento Browniano fracionário:

$$\hat{A}_{i(bff)}(t) = \bar{\rho}_i t + \bar{\psi}_i t^{\bar{H}_i} \quad (4.6)$$

onde  $\bar{\rho}_i$  e  $\bar{H}_i$  correspondem, respectivamente, à taxa média e ao parâmetro de auto-similaridade declarados pela sessão, e o parâmetro  $\bar{\psi}_i$  é equivalente ao produto  $k_i \sigma_i$  no modelo (3.13). Supondo que o tráfego da sessão  $i$  seja representado pelo tráfego Browniano fracionário  $\mathbf{A}_i(t)$ , a probabilidade de descarte de tráfego pelo algoritmo do Balde Furado Fractal é dada por:

$$\mathbb{P} \left\{ \mathbf{A}_i(t) > \hat{A}_i(t) \right\} = \overline{\mathbb{G}}(k_i)$$

que, para valores elevados de  $k_i$ , pode ser determinada utilizando o Lema (3.7). Neste caso, a escolha de valores adequados para  $k_i$  permite obter baixas probabilidades de descarte, o que comprova a eficácia deste mecanismo no policiamento de fontes auto-similares.

O algoritmo do Balde Furado Fractal é apresentado no Algoritmo 1, e seu funcionamento pode ser descrito da seguinte forma: seja uma janela de tempo de tamanho  $\Delta$ . Se o processo de chegadas excede, dentro desta janela, o valor médio declarado (dado por  $\bar{\rho}_i \Delta$ ), então todos os pacotes que violam o processo envelope  $\hat{A}_{i(bff)}(t)$  neste período são descartados, e a janela é ampliada em  $\Delta$  unidades de tempo. Esta nova janela se inicia no instante no qual o processo de chegadas viola o valor médio declarado. Este processo é, então, repetido enquanto o tráfego viola a taxa média declarada dentro da janela. No entanto, como alguns pacotes já foram descartados na janela anterior, o número de pacotes a ser descartado é, agora, igual ao número de pacotes que violaram o envelope na janela atual, menos o número de pacotes já descartados nas janelas anteriores. Quando o número médio de chegadas cai abaixo do valor declarado, o tamanho da janela é reduzido para  $\Delta$  unidades de tempo, e o processo de policiamento é reiniciado.

### 4.2.3 Uso de *buffers* com o algoritmo do Balde Furado Fractal

Na seção anterior, foi considerado que os pacotes que violam o envelope  $\hat{A}_{i(bff)}(t)$  são imediatamente descartados pelo mecanismo de policiamento. É possível, no entanto, armazená-los em um *buffer* até que eles se



---

**Algoritmo 1** Algoritmo do Balde Furado Fractal.

---

```

 $t \leftarrow 0; n \leftarrow 1;$ 
loop
  if  $A_i(t, t + n\Delta) > \bar{\rho}_i n\Delta$  then
    if  $A_i(t, t + n\Delta) > \hat{A}_{i(bff)}(t, t + n\Delta)$  then
      Descarte  $A_i(t, t + n\Delta) - \hat{A}_{i(bff)}(t, t + n\Delta) - A_i(t, t + (n - 1)\Delta) + \hat{A}_{i(bff)}(t, t + (n - 1)\Delta)$  pacotes;
    end if
     $n \leftarrow n + 1;$ 
  else
     $t \leftarrow t + n\Delta; n \leftarrow 1;$ 
  end if
end loop

```

---

tornem elegíveis para transmissão. Neste caso, uma certa quantidade de tráfego é mantida dentro do mecanismo de policiamento, introduzindo algum atraso no fluxo de pacotes. Se o tamanho do *buffer* é dado por  $b_i$ , o volume total de tráfego aceito pelo algoritmo do Balde Furado Fractal com o uso deste *buffer* seria dado por:

$$\hat{A}_{i(bff)}(t) = \bar{\rho}_i t + \bar{\psi}_i t^{\bar{H}_i} + b_i \quad (4.7)$$

Pode-se mostrar que, assintoticamente, o uso deste *buffer* tem pouca influência na probabilidade de que o tráfego gerado pela sessão viole o envelope  $\hat{A}_{i(bff)}(t)$ . De fato, assumindo que o tráfego da sessão é representado pelo processo do tráfego Browniano fracionário  $\mathbf{A}_i(t)$ , a probabilidade de violação do envelope (4.7) é dada por:

$$\begin{aligned}
P_{i(bff)}(t) &= \mathbb{P} \left\{ \mathbf{A}_i(t) > \hat{A}_{i(bff)}(t) \right\} \\
&= \mathbb{P} \left\{ \rho_i t + \sigma_i \mathbf{Z}(t) > \bar{\rho}_i t + \bar{\psi}_i t^{\bar{H}_i} + b_i \right\} \\
&= \mathbb{P} \left\{ \sigma_i t^{H_i} \mathbf{Z}(1) > (\bar{\rho}_i - \rho_i)t + \bar{\psi}_i t^{\bar{H}_i} + b_i \right\} \\
&= \mathbb{P} \left\{ \mathbf{Z}(1) > \frac{(\bar{\rho}_i - \rho_i)}{\sigma_i} t^{1-H_i} + \frac{\bar{\psi}_i}{\sigma_i} t^{\bar{H}_i-H_i} + \frac{b_i}{\sigma} t^{-H} \right\}.
\end{aligned} \quad (4.8)$$

Assim, o valor limite de  $P_{i(bff)}(t)$  quando  $t \rightarrow \infty$  é

$$\begin{aligned}
\lim_{t \rightarrow \infty} P_{i(bff)}(t) &= \exp \left\{ -\frac{1}{2\sigma_i^2} \lim_{t \rightarrow \infty} \left[ (\bar{\rho}_i - \rho_i)^2 t^{2-2H_i} + 2\bar{\psi}_i(\bar{\rho}_i - \rho_i) t^{1-2H_i+\bar{H}_i} + \bar{\psi}_i^2 t^{2\bar{H}_i-H_i} \right. \right. \\
&\quad \left. \left. + 2(\bar{\rho}_i - \rho_i)b_i t^{1-2H_i} + 2\bar{\psi}_i b_i t^{\bar{H}_i-2H_i} + b_i^2 t^{-2H_i} \right] \right\}.
\end{aligned} \quad (4.9)$$

Pode-se observar que, se  $\overline{H}_i \leq H_i$  (caso que representa vantagem para o usuário), todos os termos envolvendo  $b_i$  em (4.9) são passíveis de um rápido decaimento assintótico, de forma que têm pouca influência no valor de  $P_{i(bff)}(t)$  quando  $t \rightarrow \infty$ .

Este resultado pode ser considerado como uma consequência do fenômeno da ineficiência do *buffer*, normalmente associado ao tráfego auto-similar devido à sua característica dependência de longa duração: dado que a distribuição de probabilidade de ocupação do *buffer* decai lentamente, é provável que este *buffer* esteja com frequência completamente cheio, de modo que o tráfego correspondente à violação do envelope  $\hat{A}_{i(bff)}(t)$  será de toda forma descartado. Assim, o armazenamento de tráfego para posterior transmissão não é uma forma eficaz de reduzir o volume de dados descartado pelo policiamento. Além disso, ao adotar-se esta medida é introduzido um atraso adicional indesejado.

## Capítulo 5

# Análise da disciplina de serviço GPS

Em um modelo de serviço garantido, a implementação do controle de admissão depende da obtenção de expressões que relacionem os parâmetros de desempenho da rede ao tráfego gerado pelos usuários o que, por sua vez, depende da disciplina de serviço adotada em cada um dos nós da rede. Neste capítulo, será analisado o *Generalized Processor Sharing* (GPS), que é uma disciplina de serviço com conservação de trabalho baseada em modelos fluidos, e segundo a qual cada sessão é atendida a uma taxa proporcional a um fator de ponderação  $\phi$ .

Em geral, é bastante difícil a análise de disciplinas de serviço como o GPS, no qual o serviço destinado a cada sessão não depende exclusivamente do seu tráfego. Além disso, não é trivial a representação analítica da relação entre o serviço oferecido a uma sessão, o seu tráfego e o tráfego das demais sessões, embora seja possível provar que tal relação existe [12].

Para contornar esta dificuldade, é usual preterir a caracterização exata dos índices de desempenho em favor de expressões limitantes, matematicamente mais simples e de baixo custo computacional. Em geral, tais limitantes são obtidos para topologias de rede específicas, assumindo que o tráfego das sessões apresenta certas propriedades que simplificam a análise. Neste sentido, destacam-se os estudos de Parekh e Gallager [32, 33], que obtiveram limitantes determinísticos para o *backlog* e para o atraso correspondente a cada sessão tanto em um servidor GPS isolado como em uma rede de servidores GPS, assumindo o policiamento das sessões pelo algoritmo do Balde Furado.

Desde a publicação dos resultados de Parekh e Gallager, a análise de desempenho da disciplina GPS tem sido assunto de intensa pesquisa. Zhang, Towsley e Kurose [48] propuseram a decomposição de um servidor GPS com  $N$  sessões em um conjunto equivalente de  $N$  sistemas fila-servidor independentes, cada um alimentado por uma das sessões. A partir desta decomposição, Zhang et al. desenvolveram uma análise estocástica para a disciplina GPS, visando a obter limitantes estatísticos para o *backlog* e para o atraso de cada uma das sessões, para os quais expressões analíticas foram obtidas assumindo que o tráfego das sessões é representado por processos

de Explosividade Exponencialmente Limitada (*Exponentially Bounded Burstiness*, EBB [45, 46]). Além disso, Zhang et al. mostraram que os mesmos limitantes poderiam ser obtidos para o caso de redes de servidores GPS ao considerar que, em um servidor GPS, o processo de partidas de uma sessão qualquer é EBB se o seu processo de chegadas também é EBB.

Apesar de sua relevância, as propostas destes autores não são adequadas ao tráfego auto-similar. Embora os estudos voltados à análise da disciplina GPS sob tráfego auto-similar sejam, em sua maioria, muito recentes, alguns resultados importantes podem ser destacados. Borst, Boxma e Jelenković [3, 4] analisaram o comportamento de um servidor GPS alimentado por fontes *on/off* de cauda longa, bem como por processos de renovação com chegadas segundo uma distribuição subexponencial e tempo entre chegadas com distribuição arbitrária. Borst et al. mostraram que, em um servidor GPS, a dependência entre a taxa de serviço oferecida a uma sessão e o tráfego das demais é função apenas da taxa média destas últimas. Van Uitert e Borst [42] estenderam estes resultados para redes *feed-forward* de servidores GPS, provando que, sob determinadas condições, o desempenho desta rede é equivalente ao de uma rede *tandem* composta por dois nós, na qual cada sessão é servida a uma taxa constante, cuja dependência em relação ao tráfego das demais sessões é função apenas das suas respectivas taxas médias.

Neste capítulo, serão obtidas expressões limitantes para o atraso e o *backlog* em um servidor GPS sob tráfego auto-similar, representado matematicamente pelo processo envelope do movimento Browniano fracionário. A seguinte notação é utilizada ao longo das próximas seções: para cada sessão  $i$ , o volume de tráfego que entra no servidor no intervalo  $[\tau; t]$  é representado por  $A_i(\tau; t)$ . O volume de tráfego da sessão  $i$  que é atendido pelo servidor durante o intervalo  $[\tau; t]$  é dado por  $S_i(\tau; t)$ . De modo a simplificar a notação, o termo  $\tau$  é omitido sempre que for igual a zero. O volume de tráfego da sessão  $i$  em espera por serviço (i.e. o seu *backlog*) no instante  $t$  é dado por  $Q_i(t) = \sup_{\tau \leq t} \{A_i(\tau; t) - S_i(\tau; t)\}$ . Por definição,  $Q_i(t) = 0, \forall t \leq 0$ . O atraso introduzido no volume infinitesimal de tráfego da sessão  $i$  que chega ao servidor no instante  $t$  é dado por  $D_i(t) = \sup_{\tau \leq t} \{d(t) : A_i(t) - S_i(t + d(t)) = 0\}$ . Desta forma, o máximo *backlog* e o máximo atraso são definidos como:

$$Q_i^* = \max_t \{A_i(t) - S_i(t)\} \quad (5.1)$$

$$D_i^* = \arg \max_{d(t)} \{d(t) : A_i(t) - S_i(t + d(t)) = 0\} \quad (5.2)$$

De modo a simplificar a análise, assume-se, ao longo deste capítulo, que:

**Condição 5.1.** O tráfego de cada sessão  $i$  é policiado pelo algoritmo do Balde Furado Fractal, de modo que ele possa ser representado pelo envelope determinístico dado por (4.6), i.e.  $A_i(t) \sim \hat{A}_i(t)$ .

**Condição 5.2.** O mecanismo de policiamento descarta integralmente o tráfego de violação, e o atraso de propagação nos enlaces é desprezível. Desta forma, o *backlog* e o atraso introduzido no fluxo da sessão são devidos somente à operação do servidor GPS.

## 5.1 A disciplina de serviço GPS

O *Generalized Processor Sharing* (GPS) é uma disciplina de serviço com conservação de trabalho baseada na aproximação do fluxo de pacotes por modelos fluidos, considerando, portanto, que o tráfego gerado pelas fontes é infinitamente divisível. A Figura 5.1 apresenta o modelo básico de um servidor GPS. Neste modelo, o servidor atende ao tráfego fornecido por  $N$  fontes a uma taxa  $r$ . A cada sessão  $i$  é atribuída uma fila e um fator de ponderação  $\phi_i$ , real e positivo. O compartilhamento do servidor pelas  $N$  sessões é caracterizado pelo conjunto  $\{\phi_i\}_{1 \leq i \leq N}$ , que é denominado parametrização do servidor GPS. Em um dado instante  $\tau$ , a taxa de serviço fornecida a uma sessão  $i$  com tráfego em espera é dada por  $\frac{\phi_i}{\sum_{j \in \mathcal{B}(\tau)} \phi_j} r$ , sendo  $\mathcal{B}(\tau)$  o conjunto correspondente aos índices das sessões que apresentam tráfego em espera em  $\tau$ . Sendo  $S_i(\tau; t)$  o total de tráfego da sessão  $i$  que é servido no intervalo  $[\tau; t]$ , tem-se que [32]:

$$\frac{S_i(\tau; t)}{S_j(\tau; t)} \geq \frac{\phi_i}{\phi_j}, \quad j = 1, 2, \dots, N \quad (5.3)$$

para qualquer sessão  $i$  que apresente tráfego em espera durante todo o intervalo  $[\tau, t]$ . Desta forma, a sessão  $i$  tem garantida uma taxa de serviço dada por:

$$g_i = \frac{\phi_i}{\sum_{i=1}^N \phi_j} r, \quad i = 1, 2, \dots, N \quad (5.4)$$

Sem perda de generalidade considerar-se-á, salvo sob disposição em contrário, que a capacidade do servidor é unitária ( $r = 1$ ).

### 5.1.1 Aproximações para a disciplina GPS

Por definição, o GPS considera que o tráfego dos usuários é infinitamente divisível, ou seja, que o servidor é capaz de atender simultaneamente aos pacotes de todas as sessões. Obviamente isto não é possível em um

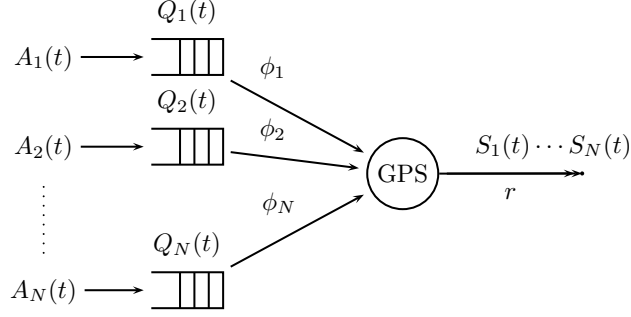


Figura 5.1: Modelo básico para um servidor GPS.

sistema prático, no qual apenas uma sessão pode receber serviço de cada vez, e um pacote deve ser inteiramente transmitido antes que seja fornecido serviço a outro. Este fato torna o GPS uma disciplina de serviço inviável do ponto de vista prático.

Diversas aproximações para o GPS têm sido implementadas em comutadores comerciais. Uma delas é denominada *Packet-by-packet Generalized Processor Sharing* (PGPS), também conhecida por *Weighted Fair Queueing* (WFQ). O princípio de funcionamento do PGPS é bastante simples: seja  $F_p$  o instante no qual o pacote  $p$  teria seu serviço completado se a disciplina de serviço fosse o GPS. O PGPS aproxima esta disciplina fornecendo serviço aos pacotes em ordem crescente de  $F_p$ , ou seja, o servidor seleciona para atendimento no instante  $\tau$  o pacote com menor  $F_p$  supondo que nenhum outro pacote chegue após  $\tau$  no servidor (considera-se que um pacote chegou ao servidor somente após a recepção do seu último bit). Uma possível implementação do PGPS é apresentada por Parekh e Gallager em [32]. Outras aproximações, de menor custo computacional, são apresentadas em [20, 47].

É importante observar que a análise desenvolvida para o GPS pode ser estendida para as suas aproximações. Neste sentido, exemplos podem ser encontrados em [20, 32, 33, 47].

## 5.2 Análise de desempenho da disciplina GPS

### 5.2.1 Sessões localmente estáveis

Como foi exposto no início deste capítulo, a análise de desempenho da disciplina GPS não é trivial. De fato, tanto (5.1) como (5.2) são de difícil solução, visto que o serviço oferecido a cada sessão  $i$  não depende somente dos seus parâmetros de tráfego, mas também do tráfego das demais sessões que são atendidas pelo servidor. Em alguns casos, porém, é possível obter limitantes conservadores de forma simples. Por definição, o servidor GPS garante a cada sessão  $i$  uma taxa de serviço mínima dada por (5.3). Desta forma, se  $g_i$  é maior que a taxa média da sessão  $i$ , dada por  $\rho_i$  no modelo (3.13), limitantes para as soluções de (5.1) e (5.2) podem ser obtidos. As sessões para as quais esta condição é satisfeita são denominadas sessões localmente estáveis, pois sua estabilidade pode ser garantida independente da estabilidade do servidor. Seja o seguinte teorema:

**Teorema 5.3.** *Seja um servidor GPS com  $N$  sessões, cujo tráfego é policiado pelo algoritmo do Balde Furado Fractal e, portanto, representado por (3.13). Se a sessão  $i$  for localmente estável,*

$$\begin{aligned} Q_i(t) &\leq Q_i^* \\ &\leq (g_i - \rho_i)^{\frac{H_i}{H_i-1}} (k_i \sigma_i)^{\frac{1}{1-H_i}} H_i^{\frac{H_i}{1-H_i}} (1 - H_i) \end{aligned} \quad (5.5)$$

*Demonstração.* A equação (5.5) é obtida diretamente de (3.18). □

Por outro lado, pode-se dizer que, sob a Condição 5.1, a solução de (5.2) requer a função inversa de  $\hat{A}_i(t)$  para  $t \geq 0$ . Embora esta função exista, ela não pode ser representada analiticamente, o que torna difícil a solução daquele problema de otimização. A alternativa proposta é o uso da aproximação [48]:

$$D_i^* \leq \frac{Q_i^*}{g_i} \quad (5.6)$$

A relação (5.6) é obtida considerando que o atraso introduzido pelo servidor no tráfego de uma sessão qualquer não pode ser maior que aquele obtido para o máximo *backlog* e taxa de serviço mínima. Desta forma, pode-se estabelecer o seguinte teorema:

**Teorema 5.4.** *Seja um servidor GPS com  $N$  sessões, cujo tráfego é policiado pelo algoritmo do Balde Furado Fractal e, portanto, representado por (3.13). Se a sessão  $i$  for localmente estável,*

$$\begin{aligned} D_i(t) &\leq D_i^* \\ &\leq \frac{1}{g_i} \left[ (g_i - \rho_i)^{\frac{H_i}{H_i-1}} (k_i \sigma_i)^{\frac{1}{1-H_i}} H_i^{\frac{H_i}{1-H_i}} (1 - H_i) \right] \end{aligned} \quad (5.7)$$

*Demonstração.* A equação (5.7) é obtida substituindo (5.5) em (5.6). □

### 5.2.2 Estabilidade de um servidor GPS

Embora a estabilidade de um servidor GPS seja garantida sempre que todas as suas sessões forem localmente estáveis, ganhos de multiplexação estatística mais significativos são obtidos quando parte das sessões não é localmente estável. Neste caso, a estabilidade destas sessões pode, sob certas condições, ser garantida diante da conservação de trabalho da disciplina GPS. Seja a seguinte definição:

**Definição 5.5.** A utilização de um servidor de capacidade  $r$  com  $N$  sessões é dada por:

$$u = \frac{1}{r} \sum_{i=1}^N \rho_i$$

sendo  $\rho_i$  a taxa média do tráfego da sessão  $i$ .

*Comentário.* Se o tráfego da sessão  $i$  é representado pelo processo envelope do movimento Browniano fracionário, então a sua taxa média é, de fato, equivalente ao parâmetro  $\rho_i$  no modelo (3.13).

A Definição 5.5 permite estabelecer o seguinte teorema:

**Teorema 5.6.** *Seja um servidor GPS compartilhado por  $N$  sessões, cujo tráfego é policiado pelo algoritmo do Balde Furado Fractal e, portanto, representado por (3.13). Se a utilização do servidor é inferior à unidade ( $u < 1$ ), então todas as sessões que dele compartilham são estáveis.*

*Demonstração.* Devido à conservação de trabalho da disciplina GPS, o total de tráfego em espera no servidor é dado por:

$$Q(t) = \sum_{j=1}^N Q_j(t) = \left( \sum_{j=1}^N \rho_j t + k_j \sigma_j t^{H_j} \right) - rt.$$

sendo  $r$  a capacidade total do servidor. No limite,

$$\begin{aligned} \lim_{t \rightarrow \infty} Q(t) &= \lim_{t \rightarrow \infty} \left( \sum_{j=1}^N \rho_j t + k_j \sigma_j t^{H_j} \right) - rt \\ &= \lim_{t \rightarrow \infty} \left( -r + \sum_{j=1}^N \rho_j + k_j \sigma_j t^{H_j-1} \right) t \\ &\simeq \lim_{t \rightarrow \infty} \left( -r + \sum_{j=1}^N \rho_j \right) t \end{aligned}$$

pois  $H_j < 1$ , para  $\forall j$ . Se  $u < 1$ , o último limite indica a ausência de tráfego em espera no servidor para  $t \rightarrow \infty$ , de modo que todas as sessões são estáveis.  $\square$

### 5.2.3 Análise de desempenho de servidores GPS pela curva de serviço

Apesar de sua simplicidade matemática, os limitantes propostos pelos Teoremas 5.3 e 5.4 são demasiado conservadores, e válidos apenas para sessões localmente estáveis. Nesta seção, é proposto um algoritmo para o



cálculo de limitantes para o *backlog* e o atraso de cada sessão em um servidor GPS sob as Condições 5.1 e 5.2. Este algoritmo é baseado na definição de uma função determinística denominada Curva de Serviço. Esta função, representada por  $\hat{S}_i(t)$ , constitui um limitante inferior para o serviço oferecido pelo GPS a uma dada sessão  $i$ .

Para um servidor GPS com  $N$  sessões, Parekh e Gallager [32] provaram que limitantes para as soluções dos problemas de otimização representados por (5.1) e (5.2) podem ser obtidos se o tráfego de todas as sessões é regulado pelo algoritmo do Balde Furado. Neste caso, os limitantes são obtidos considerando que todas as sessões são exatamente conformes desde o instante zero [32, Theorem 3].

Este resultado pode ser estendido para o caso do tráfego policiado pelo algoritmo do Balde Furado Fractal. Neste caso, para cada sessão  $i$ , é possível definir um instante de tempo  $e_i$  tal que, para todo  $t < e_i$ , a sessão possua tráfego em espera. Em  $t = e_i$ , o tráfego em espera é finalmente consumido, de modo que o intervalo  $[0; e_i]$  é denominado máximo<sup>1</sup> ciclo ativo da sessão  $i$  ou, por simplicidade, o seu ciclo ativo.

Considere os seguintes lemas, estabelecidos por Parekh e Gallager para o caso do tráfego regulado pelo algoritmo do Balde Furado, e aqui generalizados para o caso do tráfego regulado pelo algoritmo do Balde Furado Fractal [32]:

**Lema 5.7.** *Seja  $\bar{A}_1, \dots, \bar{A}_N$  o conjunto das funções que representam os processos de chegadas para os quais todas as sessões são exatamente conformes desde o instante zero. Seja ainda um instante  $t$  arbitrário que se encontra dentro do ciclo ativo da sessão  $i$  que, por sua vez, se inicia no instante  $\tau$ . Então,*

$$\bar{A}_i(t - \tau) \geq A_i(\tau; t)$$

*Demonstração.* Para o caso do tráfego regulado pelo algoritmo do Balde Furado Fractal,

$$A_i(\tau; t) \leq \rho_i(t - \tau) + \psi_i(t^{H_i} - \tau^{H_i}) \quad (5.8)$$

Note que  $t^{H_j} - \tau^{H_j} \leq (t - \tau)^{H_j}$ , pois  $H_j < 1$  para  $\forall j$ . Substituindo esta desigualdade em (5.8) obtém-se:

$$\begin{aligned} A_i(\tau; t) &\leq \rho_i(t - \tau) + \psi_i(t - \tau)^{H_i} \\ &= \bar{A}_i(t - \tau) \end{aligned}$$

□

---

<sup>1</sup>O ciclo ativo é dito máximo por ser obtido quando todas as sessões são exatamente conformes desde o instante zero.

**Lema 5.8.** *Seja  $\bar{A}_1, \dots, \bar{A}_N$  o conjunto das funções que representam os processos de chegadas para os quais todas as sessões são exatamente conformes desde o instante zero. Para toda sessão  $i$ ,  $\bar{S}_i(\tau; t)$  representa o serviço oferecido a esta sessão quando o tráfego das sessões é dado por  $\bar{A}_1, \dots, \bar{A}_N$ . Seja ainda um instante  $t$  arbitrário, que se encontra dentro do ciclo ativo da sessão  $i$  que se inicia no instante  $\tau$ . Então,*

$$\bar{S}_i(t - \tau) \leq S_i(\tau; t). \quad (5.9)$$

*Demonstração.* Se, para o conjunto de funções  $\bar{A}_1, \dots, \bar{A}_N$ ,  $\mathcal{B}$  representa o conjunto de sessões ativas no instante  $t - \tau$ , então:

$$\begin{aligned} \bar{S}_i(t - \tau) &= \frac{\phi_i}{\sum_{j \in \mathcal{B}} \phi_j} \left[ t - \tau - \sum_{j \notin \mathcal{B}} \bar{A}_j(t - \tau) \right] \\ &\leq \frac{\phi_i}{\sum_{j \in \mathcal{B}} \phi_j} \left[ t - \tau - \sum_{j \notin \mathcal{B}} A_j(\tau; t) \right] \\ &= S_i(\tau; t) \end{aligned} \quad (5.10)$$

□

**Lema 5.9.** *Para toda sessão  $i$ , os valores máximos para o tráfego em espera no servidor e o atraso por ele introduzido no fluxo de dados, respectivamente  $Q_i^*$  e  $D_i^*$ , são obtidos (não necessariamente no mesmo instante) quando todas as sessões que compartilham o servidor são reguladas pelo algoritmo do Balde Furado Fractal e exatamente conformes desde o instante zero.*

*Demonstração.* A prova aqui apresentada é semelhante àquela proposta em [32, Lemma 11]. Dado que o ciclo ativo da sessão  $i$  se inicia no instante  $\tau$ , suponha que:

$$Q_i^* = Q_i(t^*) = \max_{t \geq \tau} Q_i(t)$$

A partir dos Lemas 5.7 e 5.8 é possível verificar que,

$$\begin{aligned} \bar{A}_i(t^* - \tau) - \bar{S}_i(t^* - \tau) &\geq A_i(\tau; t^*) - S_i(\tau; t^*) \\ \bar{Q}_i(t^* - \tau) &\geq Q_i(t^*) - Q_i(\tau) \end{aligned}$$

Como, por suposição, o ciclo ativo da sessão  $i$  se inicia no instante  $\tau$ ,  $Q_i(\tau) = 0$  e  $\overline{Q}_i(t^* - \tau) \geq Q_i(t^*)$ , o que indica que o tráfego em espera é maximizado sob  $\overline{A}_1, \dots, \overline{A}_N$ . Uma prova similar pode mostrar que o atraso também é maximizado sob  $\overline{A}_1, \dots, \overline{A}_N$ .  $\square$

O Lema 5.9 é bastante intuitivo. Se uma sessão qualquer não gera o máximo de tráfego que lhe é permitido desde o instante zero, o excedente da capacidade a ela alocada é redistribuído entre as demais sessões, reduzindo assim o seu *backlog* e o seu atraso. Por outro lado, se uma sessão produz sempre o máximo de tráfego que lhe é permitido, ela impede a transmissão antecipada dos pacotes das demais sessões.

### Definição da curva de serviço

Suponha que o tráfego em espera no servidor correspondente a uma determinada sessão, arbitrariamente à Sessão 1, seja completamente consumido no instante  $t = e_1$ . O valor de  $e_1$  pode ser obtido diretamente da definição de  $Q_1(t)$ , i.e.,  $Q_1(e_1) = \hat{A}_1(e_1) - S_1(e_1) = 0$ , sendo  $S_i(t)$  o serviço oferecido à sessão  $i$  considerando que todas as sessões sejam exatamente conformes desde o instante zero, i.e.

$$S_i(t) = \frac{\phi_i}{\sum_{j=1}^N \phi_j} t$$

No instante  $t = e_2$ , o tráfego em espera de uma outra sessão,  $i = 2$ , é completamente consumido. No intervalo  $[e_1; e_2]$  o serviço oferecido a cada sessão é, portanto, descrito pela seguinte função:

$$S_i(t) = \begin{cases} \hat{A}_1(t) & , i = 1 \\ \frac{\phi_i}{\sum_{j=2}^N \phi_j} [t - e_1 - \hat{A}_1(e_1; t)] + S_i(e_1) & , i > 1 \end{cases}$$

O valor de  $e_2$  pode ser obtido considerando que  $Q_2(e_2) = \hat{A}_2(e_2) - S_2(e_2) = 0$ . Suponha que as sessões sejam ordenadas em função do instante de tempo no qual termina o seu ciclo ativo. Esta ordem pode ser considerada uma generalização daquela estabelecida por Parekh e Gallager em [32]. O serviço oferecido a qualquer sessão  $i$  durante o intervalo  $[e_{j-1}; e_j]$  é, então, dado por:

$$S_i(t) = \begin{cases} \hat{A}_i(t) & , i < j \\ \frac{\phi_i}{\sum_{l \geq j} \phi_l} [t - e_{j-1} - \sum_{l < j} \hat{A}_l(e_{j-1}; t)] + S_i(e_{j-1}) & , i \geq j \end{cases} \quad (5.11)$$

O valor de  $e_j$  pode ser obtido considerando que  $Q_j(e_j) = \hat{A}_j(e_j) - S_j(e_j) = 0$ . A curva  $\hat{S}_i(t)$  é construída a partir dos segmentos cuja inclinação e duração são dadas por:

$$s_j^i = \left. \frac{dS_i(t)}{dt} \right|_{t=e_{j-1}+} \quad (5.12)$$

$$d_j^i = e_j - e_{j-1} \quad (5.13)$$

A Figura 5.2 mostra como estes segmentos são justapostos para construir a função  $\hat{S}_i(t)$ , que corresponde a um limitante inferior e linear por partes para a curva de serviço  $S_i(t)$ . Embora a obtenção de  $\hat{S}_i(t)$  seja conceitualmente simples, deve-se observar que o cálculo de  $e_j$  não é trivial. No entanto, pode-se ainda observar que somente a duração do  $j$ -ésimo segmento de  $\hat{S}_i(t)$  depende do valor de  $e_j$ . A inclinação deste segmento depende somente de  $e_{j-1}$ , que é conhecido<sup>2</sup> desde o segmento  $j-1$ . Um limitante superior para o valor de  $e_j$  pode, então, ser obtido considerando  $\hat{A}_j(e_j) - \hat{S}_j(e_j) = 0$ . Por simplicidade, o limitante para o valor de  $e_j$  é utilizado no algoritmo computacional a ser apresentado adiante.

O número de segmentos utilizados na construção da função  $\hat{S}_i(t)$  é limitado pelo número de sessões cujo tráfego em espera pode ser completamente consumido pelo servidor, ou seja, as sessões para as quais:

$$\rho_i < \frac{\phi_i}{\sum_{l \geq i} \phi_l} \left[ 1 - \sum_{l < i} \frac{d\hat{A}_l}{dt}(e_{i-1}) \right] \quad (5.14)$$

sendo as sessões ordenadas, em relação ao índice  $i$ , de modo crescente em função do instante de tempo no qual termina o seu ciclo ativo.

Os limitantes para o máximo *backlog* e para o atraso introduzido pelo servidor no fluxo de dados da sessão  $i$  correspondem, respectivamente, às soluções dos seguintes problemas de otimização:

$$Q_i^* \leq \max_t \left\{ \hat{A}_i(t) - \hat{S}_i(t) \right\} \quad (5.15)$$

---

<sup>2</sup>De fato,  $e_j$  pode ser obtido recursivamente a partir da condição inicial  $e_0 = 0$ .

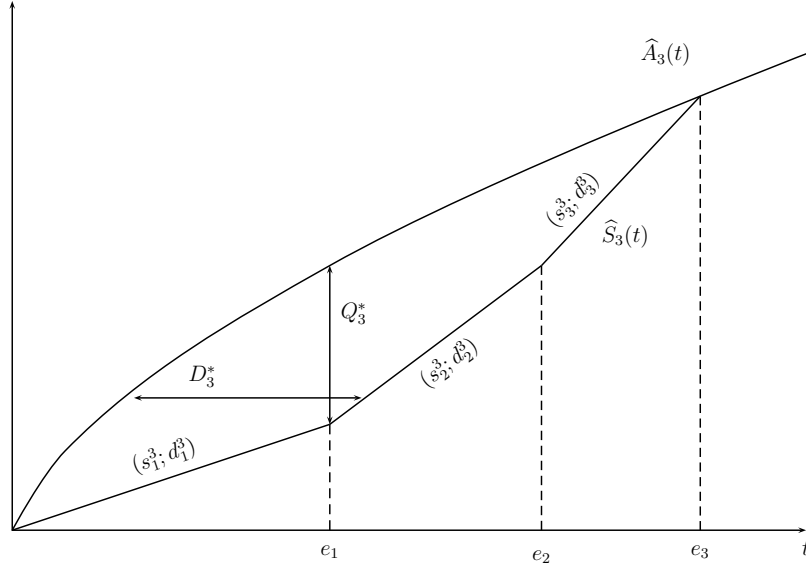


Figura 5.2: Construção da função  $\hat{S}_i(t)$  e definição de  $Q_i^*$  e  $D_i^*$  para  $i = 3$ .

$$D_i^* \leq \arg \max_{d(t)} \left\{ d(t) : \hat{A}_i(t) - \hat{S}_i(t + d(t)) = 0 \right\} \quad (5.16)$$

A Figura 5.2 também ilustra a obtenção destes limitantes. Em particular, a solução de (5.16) requer a função inversa correspondente a  $\hat{A}_i(t)$ , que não possui representação analítica. Neste caso, adota-se novamente a aproximação (5.6).

*Comentário.* Embora a condição de estabilidade apontada pelo Teorema 5.6 assegure também a existência de limitantes para todas as sessões, as diversas aproximações envolvidas na definição da curva  $\hat{S}_i(t)$  tornam possível a obtenção de limitantes somente para as sessões que satisfaçam a relação (5.14).

#### Existência e unicidade dos limitantes para $Q_i^*$ e $D_i^*$

Para o desenvolvimento de um algoritmo capaz de calcular limitantes para  $Q_i^*$  e  $D_i^*$ , é necessário provar que o problema de otimização representado por (5.15) pode ser resolvido quando a relação (5.14) é satisfeita. Seja o seguinte teorema:

**Teorema 5.10.** *Seja uma sessão  $i$ , exatamente conforme desde o instante zero ao processo envelope  $\hat{A}_i(t)$ , dado por (3.13). O problema de otimização dado pela relação (5.15) pode ser resolvido sempre que (5.14) for válida.*

*Demonstração.* Por definição,  $\hat{S}_i(t) < \hat{A}_i(t)$  para  $t \in (0; e_i)$ . A condição de igualdade  $\hat{S}_i(t) = \hat{A}_i(t)$  é atingida em  $t = 0$  e  $t = e_i$ . Portanto,  $Q_i(0) = Q_i(e_i) = 0$  e  $Q_i(t) > 0$  para  $t \in (0; e_i)$ . Para mostrar que existe no mínimo

uma solução para (5.15), é suficiente mostrar que  $e_i$  é finito. Dado que a função  $\hat{S}_i(t)$  é estritamente côncava no intervalo  $[0; e_i]$ ,  $e_i$  é finito se, para qualquer  $j < i$ ,

$$\rho_i < s_j^i$$

Assim, se a taxa de serviço oferecida à sessão  $i$  nunca for maior que sua taxa média,  $\hat{S}_i(t)$  nunca atinge  $\hat{A}_i(t)$  e o volume de tráfego em espera cresce indefinidamente. No entanto, se em algum momento, esta sessão é servida a uma taxa maior que a taxa média, esta situação permanecerá até que  $\hat{S}_i(t)$  atinja  $\hat{A}_i(t)$ , definindo assim um instante  $e_i$  finito no qual o tráfego em espera é completamente consumido. É possível concluir, portanto, que a condição suficiente para a existência de um segmento cuja inclinação  $s_j^i$  seja maior que  $\rho_i$  é dada por (5.14); se esta condição for satisfeita, no mínimo uma solução para (5.15) pode ser obtida.  $\square$

Além da existência, é possível assegurar a unicidade da solução. Seja o seguinte lema:

**Lema 5.11.** *Se  $f(x)$  e  $g(x)$  são, respectivamente, uma função estritamente convexa e uma função estritamente côncava no intervalo  $[x_1; x_2]$ , a função  $h(x) = f(x) - g(x)$  é estritamente convexa no mesmo intervalo.*

Como é garantida a unicidade do máximo de uma função estritamente convexa, o seguinte teorema pode ser estabelecido:

**Teorema 5.12.** *Seja uma sessão  $i$ , exatamente conforme desde o instante zero ao processo envelope  $\hat{A}_i(t)$ , dado por (3.13). Se o problema de otimização dado por (5.15) for passível de solução, então sua solução é única.*

*Demonstração.* Sendo a função  $\hat{A}_i(t)$  estritamente convexa e a função  $\hat{S}_i(t)$  estritamente côncava no intervalo  $[0; e_i]$ , a diferença  $\hat{A}_i(t) - \hat{S}_i(t)$  é estritamente convexa neste intervalo. A solução de (5.15) é, portanto, única.  $\square$

Resultados equivalentes podem ser obtidos para provar a existência e unicidade de  $D_i^*$  quando calculado por (5.16). Entretanto, sendo o cálculo realizado por meio da aproximação (5.6), pode-se afirmar que é suficiente provar a existência e unicidade de  $Q_i^*$  para garantir as mesmas propriedades para  $D_i^*$ . Observa-se ainda que os resultados apontados pelos Teoremas 5.10 e 5.12 indicam que a obtenção de limitantes depende dos parâmetros de tráfego das sessões e da parametrização GPS do servidor, mas não depende do número de sessões atendidas pelo servidor.

### Caracterização do processo de partidas

A análise de redes de servidores GPS requer a caracterização adequada da função  $S_i(t)$ , que descreve o volume de tráfego da sessão  $i$  servido até o instante  $t$ . Parekh e Gallager [32] caracterizam  $S_i(t)$  utilizando um descritor

de tráfego sugerido por Cruz [9, 10]. No entanto, esta caracterização não é adequada ao tráfego auto-similar, dado que ela assume que  $S_i(t)$  pode ser representado por uma função linear no tempo.

Assim, é proposta a seguinte caracterização de  $S_i(t)$ :

$$S_i(t) \sim \hat{A}_i(t) \quad (5.17)$$

i.e., se o processo envelope  $\hat{A}_i(t)$  restringe o processo de chegadas da sessão  $i$ , ele também restringe o correspondente processo de partidas,  $S_i(t)$ . De fato, se uma sessão  $i$  satisfaz a relação (5.14), existe um instante  $e_i$  tal que  $S_i(t) < \hat{A}_i(t)$  para  $t < e_i$  e  $S_i(t) = \hat{A}_i(t)$  para  $t \geq e_i$ . O envelope mais justo que pode ser obtido para caracterizar  $S_i(t)$  para  $\forall t > 0$  é, portanto,  $S_i(t) = \hat{A}_i(t)$ . Por outro lado, se a relação (5.14) não é satisfeita, então  $S_i(t) < \hat{A}_i(t)$  para todo  $t > 0$ , o que significa que  $\hat{A}_i(t)$  continua sendo um envelope válido, embora não seja o mais justo possível.

#### Algoritmo computacional para o cálculo de $Q_i^*$ e $D_i^*$

O algoritmo computacional para a obtenção de limitantes para o *backlog* e para o atraso introduzido por um servidor GPS isolado é apresentado no Algoritmo 2. Este algoritmo é capaz de obter limitantes para todas as sessões que satisfaçam (5.14), e não requer que as sessões sejam previamente ordenadas de acordo com o critério proposto em [32].

O bloco principal do Algoritmo 2 funciona da seguinte maneira: primeiro, a inclinação do  $j$ -ésimo segmento da curva  $\hat{S}_i(t)$ , representado por  $s_j^i$ , é calculado para todas as sessões  $i$ . O cálculo de  $s_j^i$  pode ser realizado de duas maneiras, em função da sessão  $i$  estar ou não em seu ciclo ativo. Define-se, então, um conjunto de sessões  $i$  estáveis (i.e. para as quais  $s_j^i > \rho_i$ ) e para as quais  $Q_i^*$  e  $D_i^*$  ainda não tenham sido calculados. A sessão que primeiro termina o seu ciclo ativo é, então, identificada. Esta informação é utilizada para determinar o instante  $e_j$ , no qual o  $j$ -ésimo segmento de  $\hat{S}_i(t)$  termina. Finalmente, para todas as sessões cujo volume máximo de tráfego em espera é atingido durante o  $j$ -ésimo segmento, os limitantes são calculados.

O algoritmo termina quando os limitantes forem calculados para todas as sessões que satisfaçam a relação (5.14).

---

**Algoritmo 2** Algoritmo para o cálculo de  $Q_i^*$  e  $D_i^*$ .

---

```

 $e_0 \leftarrow 0;$ 
for  $\forall i, \gamma_i \leftarrow 0;$ 
 $j \leftarrow 1;$ 
 $\mathbf{E} \leftarrow$  Conjunto de todas as sessões;
{Calcula  $s_j^i$  para toda sessão  $i$ .}
loop
  for  $\forall i \notin \mathbf{E}, s_j^i \leftarrow \rho_i + H_i \psi_i e_{j-1}^{H_i-1};$ 
  for  $\forall i \in \mathbf{E}, s_j^i \leftarrow \frac{\phi_i}{\sum_{k \in \mathbf{E}} \phi_k} (1 - \sum_{k \notin \mathbf{E}} s_k^i);$ 
   $\mathbf{C} \leftarrow \{i \in \mathbf{E} \mid s_j^i > \rho_i\}$ 
  if  $\mathbf{C} = \emptyset$  then
    break; {Para as demais sessões não podem ser obtidos limitantes.}
  end if
  {Determine qual sessão termina seu ciclo ativo primeiro.}
  for  $\forall i \in \mathbf{C}$  do
    Determine  $t_i$  tal que  $(\rho_i - s_j^i) t_i + s_j^i e_{j-1} - \gamma_i + \psi_i t_i^{H_i} = 0;$ 
  end for
   $e_j \leftarrow \min_i t_i;$ 
   $\mathbf{E} \leftarrow \mathbf{E} - \arg \min_i t_i$ 
  {Para todo  $i \in \mathbf{C}$ , determine  $Q_i^*$  e  $D_i^*$  se possível. }
  for  $\forall i \in \mathbf{C}$  do
     $t^* \leftarrow \left( \frac{s_j^i - \rho_i}{\psi_i H_i} \right)^{\frac{1}{H_i-1}};$ 
     $t^* \leftarrow \max(t^*, e_{j-1});$ 
    {Calcule  $Q_i^*$  e  $D_i^*$ , se  $t^* \in [e_{j-1}, e_j]$ . }
    if  $t^* < e_j$  then
       $Q_i^* \leftarrow (\rho_i - s_j^i) t^* - \gamma_i + s_j^i e_{j-1} + \psi_i t^{*H_i};$ 
       $D_i^* \leftarrow Q_i^* \left( \frac{\phi_i}{\sum_{k=1}^N \phi_k} \right)^{-1};$ 
    end if
     $\gamma_i \leftarrow \gamma_i + s_j^i (e_j - e_{j-1});$ 
  end for
   $j \leftarrow j + 1;$ 
end loop

```

---



### 5.3 Exemplo numérico

Nesta seção, o cálculo de limitantes de atraso por meio do Algoritmo 2 é ilustrado. A precisão dos valores obtidos é verificada por meio de comparação com resultados obtidos por simulação.

As Figuras 5.3 a 5.5, mostram os limitantes de atraso correspondente a três sessões, em função da utilização do servidor. Os parâmetros de tráfego e os fatores de ponderação do servidor GPS são mostrados na Tabela 5.1. Os resultados de simulação foram obtidos utilizando o pacote Matlab/Simulink, no qual foi implementado um servidor GPS alimentado por séries temporais com  $10^6$  amostras e exatamente conformes desde o instante zero. Os limitantes obtidos a partir da aproximação por serviço de taxa constante, que são mostrados nas mesmas figuras, foram obtidos assumindo que cada sessão é servida à taxa mínima que lhe é garantida pelo servidor GPS (Teorema 5.4). Estes limitantes, de caráter conservador, são apresentados para fins de comparação.

Dados os parâmetros de tráfego e os fatores de ponderação mostrados na Tabela 5.1, a sessão que encerra primeiro o seu ciclo ativo é a Sessão 3. Como todas as sessões são exatamente conformes desde o instante zero, todas estão ativas durante o ciclo ativo da Sessão 3. Assim, até o fim do seu ciclo ativo, esta sessão é efetivamente servida à taxa mínima garantida pelo GPS. Por esta razão, os limitantes correspondentes à aproximação por serviço de taxa constante e à análise da curva de serviço (esta dada pelo Algoritmo 2) são coincidentes.

Após o fim do ciclo ativo da Sessão 3, a próxima sessão a terminar o seu ciclo ativo é a Sessão 2. Para esta sessão, pode-se verificar que o limitante correspondente à análise da curva de serviço é mais próximo do resultado das simulações que aquele obtido por meio da aproximação por serviço de taxa constante. Isto é ainda mais óbvio no caso de altos valores de utilização (onde os ganhos de multiplexação estatística são mais relevantes) e no caso da Sessão 1, que é a última a terminar o seu ciclo ativo. A diferença entre os limitantes obtidos por meio da análise da Curva de Serviço e por meio de simulação se deve às aproximações adotadas na obtenção da curva  $\hat{S}_i(t)$ .

Para os mesmos parâmetros de tráfego, resultados similares foram obtidos utilizando seqüências fBm com  $10^6$  amostras e geradas utilizando o método de Mandelbrot [5] (Figuras 5.6 a 5.8). É possível observar que os limitantes obtidos continuam válidos, embora não sejam tão precisos como no exemplo anterior, pelo fato das seqüências utilizadas não serem exatamente conformes.

Os resultados apresentados comprovam a eficácia do Algoritmo 2 na caracterização do atraso introduzido por um servidor GPS em um fluxo de tráfego auto-similar.

Sessão	$\rho_i$	$\sigma_i$	$k_i$	$H_i$	$\phi_i$
1	0,20	0,15	5,25	0,85	0,215
2	0,40	0,30	5,25	0,75	0,430
3	0,30	0,20	5,25	0,70	0,355

Tabela 5.1: Parâmetros de tráfego e do servidor GPS utilizados no exemplo.

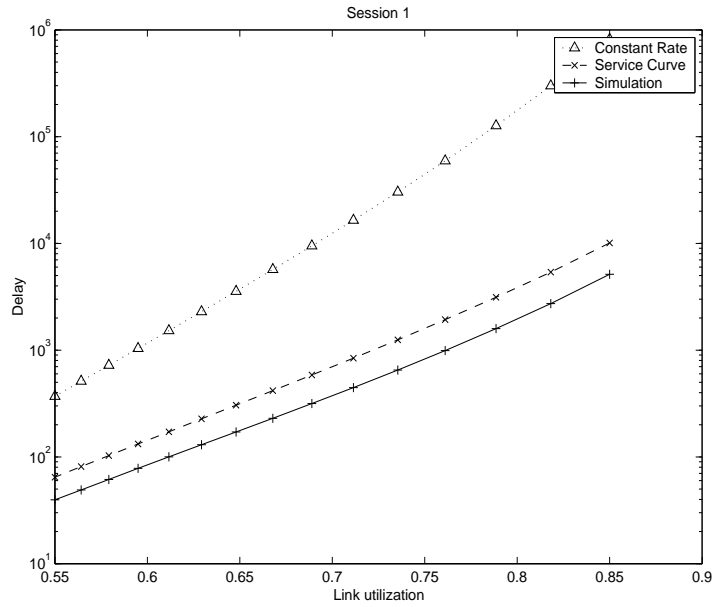


Figura 5.3: Resultados obtidos para a Sessão 1, sob tráfego exatamente conforme desde o instante zero.

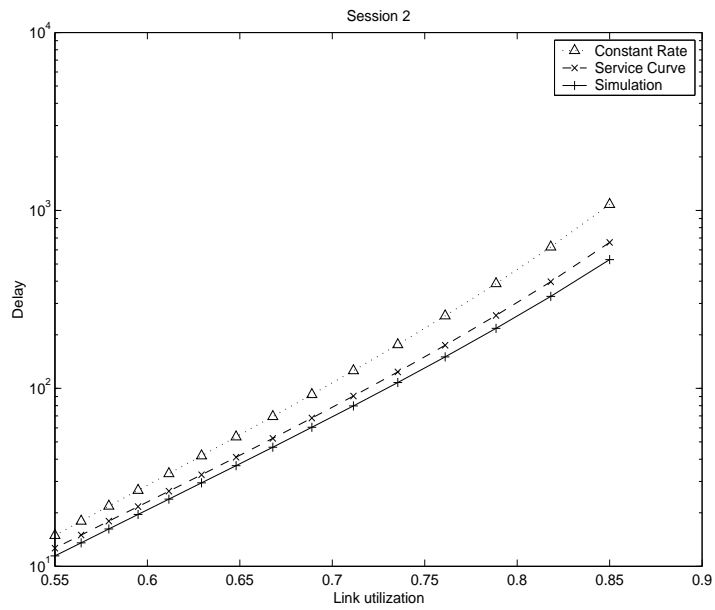


Figura 5.4: Resultados obtidos para a Sessão 2, sob tráfego exatamente conforme desde o instante zero.

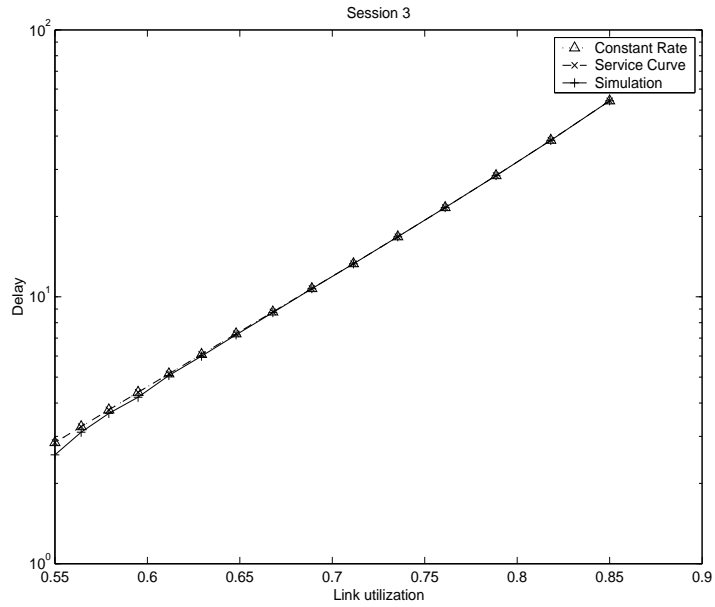


Figura 5.5: Resultados obtidos para a Sessão 3, sob tráfego exatamente conforme desde o instante zero.

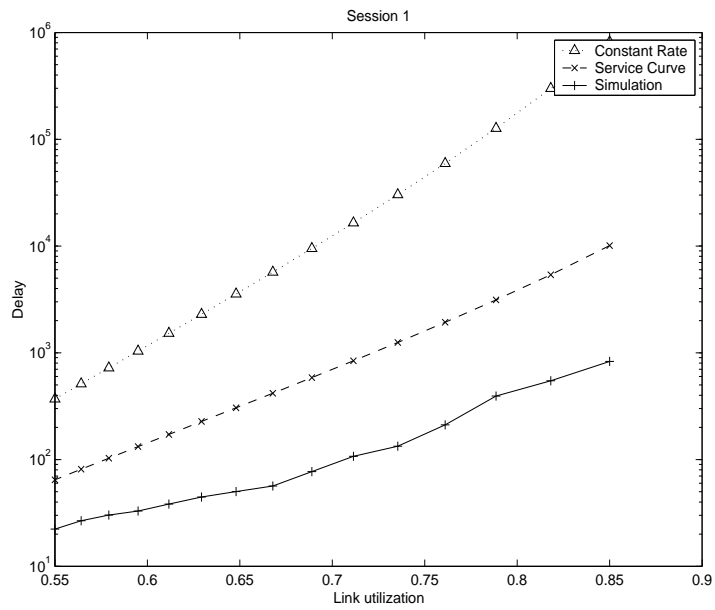


Figura 5.6: Resultados obtidos para a Sessão 1, quando o tráfego das sessões é representado por seqüências fBm.

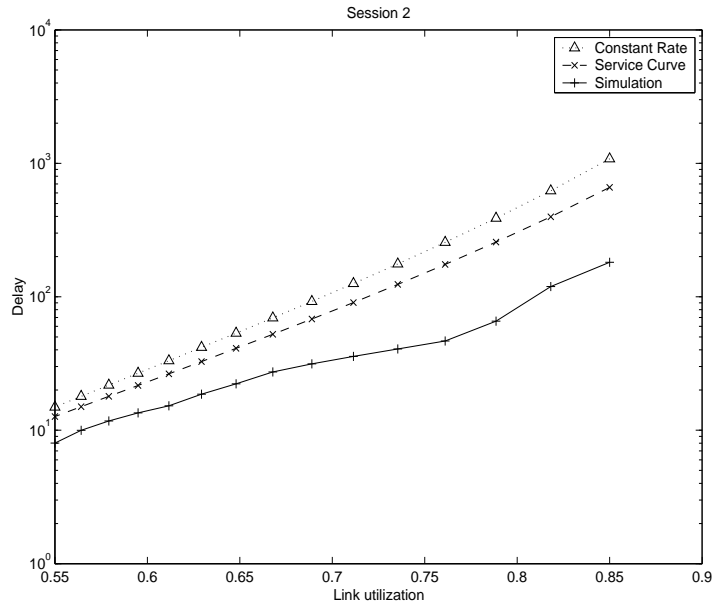


Figura 5.7: Resultados obtidos para a Sessão 2, quando o tráfego das sessões é representado por seqüências fBm.

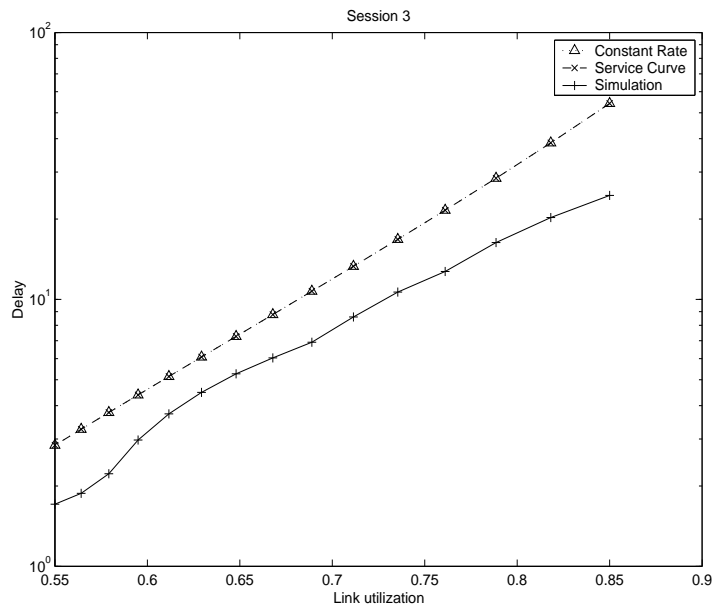


Figura 5.8: Resultados obtidos para a Sessão 3, quando o tráfego das sessões é representado por seqüências fBm.

## Capítulo 6

# Análise de desempenho de redes de servidores GPS

### 6.1 Modelo de rede

Nesta seção, será apresentada uma técnica de análise de desempenho para uma rede de servidores GPS com topologia arbitrária. Considera-se que esta rede pode ser representada como um grafo orientado, cujos nós representam comutadores e cujos arcos representam enlaces. Cada comutador é composto de um servidor GPS de capacidade  $r^m$  (onde  $m$  representa o  $m$ -ésimo servidor), e um conjunto de filas independentes, uma atribuída a cada sessão atendida por este servidor. Seja  $N$  o número de sessões que chegam à rede e  $n_m$  o número de sessões servidas pelo  $m$ -ésimo servidor. O caminho utilizado pela sessão  $i$  é simbolizado por  $P(i)$ , e  $K_i$  é o número total de nós neste caminho.

Para cada sessão  $i$ , a quantidade de tráfego que entra na rede durante o intervalo  $[\tau; t]$  é dado por  $A_i(\tau; t)$ . A quantidade de tráfego da sessão  $i$  servido pelo  $k$ -ésimo nó em  $P(i)$  durante o intervalo  $[\tau; t]$  é representado por processo  $S_i^{(k)}(\tau; t)$ ,  $k = 1, \dots, K_i$ . De modo a simplificar a notação,  $\tau$  será omitido sempre que for igual a zero.

O conjunto de sessões atendidas pelo servidor  $m$  é representado por  $I(m)$ . Para toda sessão  $i \in I(m)$ ,  $A_i^m(\tau; t)$  representa a quantidade de tráfego da sessão  $i$  que chega ao servidor  $m$  durante o intervalo  $[\tau; t]$ , e  $S_i^m(\tau; t)$  é a quantidade que o deixa no mesmo intervalo.

Além das Condições 5.1 e 5.2 estabelecidas no capítulo anterior, assume-se que :

**Condição 6.1.** Para toda sessão  $i$ , o tráfego é restrito ao caminho  $P(i)$  previamente estabelecido;

**Condição 6.2.** O atraso de propagação nos enlaces da rede é desprezível, de modo que o *backlog* e o atraso introduzido no fluxo das sessões são devidos apenas à operação dos servidores GPS.

Para que seja possível analisar uma rede de servidores GPS, também é necessário caracterizar  $A_i^m(t)$ . Na Seção 5.2 foi mostrado que  $S_i^m(t)$  pode ser representado pelo processo envelope associado a  $A_i^m(t)$ . Se este procedimento for aplicado de modo recursivo a todos os nós ao longo do caminho  $P(i)$ , pode-se estabelecer o seguinte lema:

**Lema 6.3.** *Se o processo envelope  $\hat{A}_i(t)$ , dado por (3.13), representa o tráfego da sessão  $i$  que entra na rede, então ele também representa o tráfego da sessão  $i$  que entra no servidor  $m \in P(i)$ , i.e.  $A_i^m(t) \sim \hat{A}_i(t)$ .*

Finalmente, é necessário definir o atraso fim-a-fim e *backlog* correspondentes a cada sessão. Seja  $Q_i^m(t) = A_i^m(t) - S_i^m(t)$  a realização do processo *backlog* correspondente à sessão  $i$  no servidor  $m \in P(i)$ . Por definição,  $Q_i^m(t) = 0, \forall t \leq 0$ . O *backlog* total da sessão  $i$  é, então, dado por  $Q_i(t) = \sum_{m \in P(i)} Q_i^m(t)$ , e  $D_i(t) = \{d(t) : A_i(t) - S_i^{(K_i)}(t + d(t)) = 0\}$  corresponde ao atraso introduzido pela rede no tráfego da sessão  $i$  que entra na rede no instante  $t$ . Desta forma, o *máximo backlog* e o máximo atraso fim-a-fim são definidos como:

$$Q_i^* = \max_t \sum_{m \in P(i)} [A_i^m(t) - S_i^m(t)] \quad (6.1)$$

$$D_i^* = \arg \max_{d(t)} \left\{ d(t) : A_i(t) - S_i^{(K_i)}(t + d(t)) = 0 \right\} \quad (6.2)$$

## 6.2 Estabilidade de redes de servidores GPS sob tráfego auto-similar

### 6.2.1 Sessões localmente estáveis

Como no caso de um servidor GPS isolado, os problemas de otimização representados por (6.1) e (6.2) são de difícil solução para uma rede de servidores GPS com topologia arbitrária. Em alguns casos, entretanto, é possível obter limitantes conservadores de forma simples. Por definição, o servidor  $m$ , garante a cada sessão  $i \in I(m)$  uma taxa de serviço mínima dada por:

$$g_i^m = \frac{\phi_i^m}{\sum_{j \in I(m)} \phi_j^m} r^m$$

onde  $\phi_i^m$  é o fator de ponderação GPS associado à sessão  $i$  no servidor  $m$ . Desta forma, pode-se estabelecer que a taxa de serviço mínima garantida à sessão  $i$  ao longo do caminho  $P(i)$  é:

$$g_i = \min_{m \in P(i)} g_i^m \quad (6.3)$$

Em [33] foi mostrado que, se uma sessão  $i$  é regulada pelo algoritmo do Balde Furado e  $g_i$  é maior que a taxa média daquela sessão, limitantes para as soluções de (6.1) e (6.2) podem ser obtidos. Como no caso de um servidor GPS isolado, as sessões para as quais estas condições são satisfeitas também são denominadas localmente estáveis. Resultados similares podem ser obtidos quando o tráfego das sessões é regulado pelo algoritmo do Balde Furado Fractal, como é provado no seguinte teorema:

**Teorema 6.4.** *Para uma sessão  $i$  qualquer que seja regulada pelo algoritmo do Balde Furado Fractal, e para a qual  $g_i > \rho_i$  :*

$$\begin{aligned} Q_i(t) &\leq Q_i^* \\ &\leq (g_i - \rho_i)^{\frac{H_i}{H_i-1}} (k_i \sigma_i)^{\frac{1}{1-H_i}} H_i^{\frac{H_i}{1-H_i}} (1 - H_i) \end{aligned} \quad (6.4)$$

$$\begin{aligned} D_i(t) &\leq D_i^* \\ &\leq \frac{1}{g_i} \left[ (g_i - \rho_i)^{\frac{H_i}{H_i-1}} (k_i \sigma_i)^{\frac{1}{1-H_i}} H_i^{\frac{H_i}{1-H_i}} (1 - H_i) \right] \end{aligned} \quad (6.5)$$

*Demonstração.* As equações (6.4) e (6.5) podem ser obtidas diretamente de (3.18) e (5.6), respectivamente.  $\square$

### Estabilidade da rede na presença de sessões que não são localmente estáveis

Sempre que todas as sessões que utilizem uma rede forem localmente estáveis, a estabilidade da rede inteira pode ser assegurada. No entanto, como foi argumentado no capítulo anterior, em algumas situações pode ser interessante que apenas algumas sessões sejam localmente estáveis. Sabe-se que a presença de sessões que não sejam localmente estáveis pode causar realimentações implícitas que levem a rede à instabilidade, e que tal efeito é devido ao tratamento inconsistente das sessões dentro da rede [33]. Desta forma, para que a estabilidade da rede seja assegurada, é necessário estabelecer as condições sob as quais evita-se tal tratamento inconsistente.

Supondo que todas as sessões sejam reguladas pelo algoritmo do Balde Furado, Parekh e Gallager [33] demonstraram que as condições de estabilidade para redes de servidores GPS podem ser estabelecidas para uma classe de parametrização GPS denominada Tratamento de Sessões Relativamente Consistente (*Consistent Relative Session*

*Treatment*, CRST). É possível generalizar este resultado para o caso do tráfego regulado pelo algoritmo do Balde Furado Fractal. Considere as seguintes definições, e o seguinte teorema [33]:

**Definição 6.5.** Uma sessão  $j$  é dita impeditiva para uma outra sessão  $i$ , no servidor  $m$ , se

$$\frac{\phi_i^m}{\phi_j^m} < \frac{\rho_i}{\rho_j}$$

**Definição 6.6.** O Tratamento de Sessões Relativamente Consistente (*Consistent Relative Session Treatment*, CRST) é uma parametrização GPS para a qual existe uma ordem entre as sessões tal que, para duas sessões  $i, j$  quaisquer, se a sessão  $i$  antecede a sessão  $j$  na ordem, então a sessão  $i$  não impede a sessão  $j$  em nenhum servidor da rede.

**Teorema 6.7.** *Suponha que as sessões  $i$  e  $j$  compartilhem um servidor  $m$ , e que a sessão  $j$  seja estável. Se esta sessão  $j$  não é impeditiva para a sessão  $i$  naquele servidor, então o tráfego da sessão  $j$  não pode levar a sessão  $i$  à instabilidade naquele servidor.*

*Demonstração.* Seja  $\tau_i^m$  o instante de tempo no qual o *backlog* da sessão  $i$  atinge o seu valor máximo, e  $e_i^m$  o instante no qual ele é completamente consumido, conforme definido na Seção 5.2.3. Obviamente,  $e_i^m \geq \tau_i^m$ . Se  $e_j^m$  representa o tempo no qual o *backlog* da sessão  $j$  é completamente consumido, há duas possibilidades:

1.  $e_i^m \leq e_j^m$ .

Neste caso, a sessão  $j$  termina o seu ciclo ativo depois da sessão  $i$ . Considerando que as sessões são ordenadas segundo o término do seu ciclo ativo, a sessão  $i$  antecede a sessão  $j$ , então:

$$S_i^m(e_i^m) = \frac{\phi_i^m}{\sum_{l \geq i} \phi_l^m} \left[ e_i^m - \sum_{l < i} S_l^m(e_i^m) \right]$$

Assim, a quantidade de serviço oferecido à sessão  $i$  depende somente da parametrização GPS e do tráfego das sessões que a antecedem na ordem. Desta forma, a estabilidade da sessão  $i$  não depende diretamente do tráfego da sessão  $j$ .

2.  $e_i^m > e_j^m$ .

Neste caso, a sessão  $j$  termina o seu ciclo ativo antes da sessão  $i$ . Considerando que o pior caso para estabilidade é obtido quando todas as sessões são exatamente conformes desde o instante zero, e dado que a quantidade total de serviço oferecido à sessão  $i$  depende, neste caso, de todas as sessões que terminam



o seu ciclo ativo antes da própria sessão  $i$ , é possível que esta sessão seja instável no servidor  $m$ , o que eventualmente depende do tráfego da sessão  $j$ . No entanto, como é suposto que a sessão  $j$  não é impeditiva para a sessão  $i$  no servidor  $m$ , é possível mostrar que a estabilidade da sessão  $i$  não depende do tráfego da sessão  $j$ . Se a sessão  $j$  é estável, então

$$S_j^m(0; e_j^m) > \rho_j e_j^m \geq \frac{\rho_i \phi_j^m}{\phi_i^m} e_j^m \quad (6.6)$$

Esta desigualdade é devida ao fato da sessão  $j$  não ser impeditiva para a sessão  $i$ . Substituindo (6.6) no lugar de  $S_j(\cdot)$  em (5.3),

$$S_i^m(0; e_j^m) = \frac{\phi_i^m}{\phi_j^m} S_j^m(0; e_j^m) > \rho_i e_j^m$$

Desta forma, a estabilidade da sessão  $i$  não depende do tráfego da sessão  $j$  — se a sessão  $i$  é, de fato, instável, sua instabilidade não é devida ao tráfego da sessão  $j$ .

□

A partir do Teorema 6.7, é possível concluir que somente as sessões que são impeditivas para a sessão  $i$  podem torná-la instável no servidor  $m \in P(i)$ . Fica claro, portanto, que a sessão  $i$  sofreria realimentação se uma sessão  $j$  fosse para ela impeditiva em um servidor  $m_1$  e justamente o contrário se verificasse em um outro servidor  $m_2$ . Neste caso, a quantidade de serviço oferecido à sessão  $i$  dependeria daquele oferecido à sessão  $j$ , que, por sua vez, dependeria do serviço oferecido à própria sessão  $i$  em outro servidor. Este efeito pode ser completamente eliminado com o uso de uma parametrização CRST — neste caso, a estabilidade de uma sessão depende apenas do tráfego das sessões que são para ela impeditivas em algum dos servidores ao longo do seu caminho na rede.

Resta ainda estabelecer uma condição suficiente para garantir a estabilidade da rede na presença de sessões que não sejam localmente estáveis. Segundo a Definição 5.5, a utilização do servidor  $m$  é dada por:

$$u^m = \frac{1}{r^m} \sum_{j \in I(m)} \rho_j$$

**Teorema 6.8.** *Se  $u^m < 1$  e se todas as sessões são reguladas pelo algoritmo do Balde Furado Fractal na entrada da rede, então todas as sessões que compartilham o servidor  $m$  são estáveis.*

*Demonstração.* Ver Teorema 5.6. □

**Lema 6.9.** *Se uma rede de servidores GPS opera sob parametrização CRST, uma sessão  $i$  será estável se  $u^m < 1$ ,  $\forall m \in P(i)$ .*

Estes resultados confirmam a validade do seguinte corolário<sup>1</sup> para o caso do tráfego regulado pelo algoritmo do Balde Furado Fractal:

**Corolário 6.10.** *Uma rede de servidores GPS que opere sob parametrização CRST será estável se  $u^m < 1$ ,  $\forall m$ .*

### 6.2.2 Cálculo do backlog e do atraso fim-a-fim das sessões em uma rede de servidores GPS

Seja uma rede de servidores GPS-CRST estável, cujas sessões sejam reguladas pelo algoritmo do Balde Furado Fractal. Pelo Teorema 6.4, limitantes para o *backlog* e para o atraso fim-a-fim foram estabelecidos para as sessões denominadas localmente estáveis. No entanto, estes limitantes não levam em consideração a interação entre as sessões, tampouco são aplicáveis a sessões que não sejam localmente estáveis. Parekh e Gallager [33] apresentaram um algoritmo para o cálculo de limitantes para o *backlog* e o atraso fim-a-fim de cada uma das sessões em uma rede GPS-CRST estável. Embora este algoritmo tenha sido obtido considerando que todas as sessões são reguladas pelo algoritmo do Balde Furado, um algoritmo equivalente pode ser obtido para o caso do tráfego regulado pelo algoritmo do Balde Furado Fractal.

Por simplicidade, apenas uma única sessão  $i$  será considerada. O caminho percorrido pelo tráfego desta sessão é  $P(i) = \{1, 2, \dots, K\}$ , e o processo envelope a ela associado é dado por  $\hat{A}_i(t)$ . Segundo o Lema 6.3,  $A_i^m(t) \sim \hat{A}_i(t)$  para todo  $m \in P(i)$ . Além disso, assume-se que [33]:

**Condição 6.11.** Para  $\forall m \in P(i)$ , toda sessão  $j \in I(m) - \{i\}$  é livre para gerar tráfego, desde que  $A_j^m(t) \sim \hat{A}_j(t)$ .

**Condição 6.12.** O tráfego da sessão  $i$  é restrito ao caminho  $P(i)$ .

Os valores de  $D_i^*$  e  $Q_i^*$  obtidos sob estas condições, que garantem a independência entre as sessões, são limitantes para as soluções exatas de (6.1) e (6.2). Parekh e Gallager [33] demonstraram que tais limitantes podem ser obtidos considerando que o tráfego segue um padrão tal que as sessões se tornam exatamente conformes no servidor  $m$  de forma simultânea, mas somente depois que as sessões no servidor  $m - 1$  tornem-se exatamente conformes. Adicionalmente, foi mostrado que os limitantes para  $D_i^*$  e  $Q_i^*$  podem ser obtidos sob padrões de chegadas diferentes [33, Fig. 4].

De modo a evitar complicações decorrentes da multiplicidade de padrões de tráfego para a obtenção dos limitantes, Parekh e Gallager [33] propõem o uso de uma única função, denominada Curva de Serviço Universal, a partir da qual limitantes para  $Q_i^*$  e  $D_i^*$  podem ser obtidos sob as Condições 6.11 e 6.12.

<sup>1</sup>Este corolário foi estabelecido por Parekh e Gallager [33] para o caso do tráfego regulado pelo algoritmo do Balde Furado.

O mesmo raciocínio pode ser adotado quando todas as sessões são policiadas pelo algoritmo do Balde Furado Fractal, como será mostrado adiante. Neste caso, entretanto, somente os limitantes para  $Q_i^*$  são obtidos a partir da Curva de Serviço Universal. Devido à falta de representação analítica para a inversa de  $\hat{A}_i(t)$ , os limitantes para  $D_i^*$  são obtidos por meio de (5.6), sendo  $g_i$  dado por (6.3).

## A Curva de Serviço Universal

A Curva de Serviço Universal para a sessão  $i$  é obtida da seguinte forma [33]: seja  $\hat{S}_i^m$  a curva de serviço correspondente à sessão  $i$  no servidor  $m \in P(i)$ . Segundo o Lema 6.3, esta curva é obtida por meio da análise proposta na Seção 5.2.3, considerando que  $A_i^m(t) \sim \hat{A}_i(t)$ . Para cada servidor  $m \in P(i)$ ,  $\hat{S}_i^m$  é uma função contínua, linear por partes e côncava no intervalo  $[0; e_i^m]$ , sendo  $e_i^m$  a duração do ciclo ativo da sessão  $i$  naquele servidor. Sendo a inclinação e a duração do  $j$ -ésimo segmento de  $\hat{S}_i^m$  representadas, respectivamente, por  $s_j^{i,m}$  e  $d_j^{i,m}$ , um conjunto  $E_i^m$  pode ser definido como:

$$E_i^m = \left\{ \left( s_1^{i,m}; d_1^{i,m} \right) \left( s_2^{i,m}; d_2^{i,m} \right) \dots \left( s_{n_{i,m}}^{i,m}; d_{n_{i,m}}^{i,m} \right) \right\}$$

onde  $n_{i,m}$  corresponde ao número de segmentos dentro do intervalo  $[0; e_i^m]$ . Seja  $E_i$  o conjunto união dos conjuntos  $E_i^m$  para todo  $m \in P(i)$ :

$$E_i = \bigcup_{m \in P(i)} \bigcup_{j=1}^{n_m} \left\{ \left( s_j^{i,m}; d_j^{i,m} \right) \right\} \quad (6.7)$$

Seja  $G_i(t)$  a curva linear por partes obtida por meio da justaposição dos elementos de  $E_i$  em ordem crescente de inclinação. A Curva de Serviço Universal para a sessão  $i$  é, então, definida como [33]:

$$U_i(t) = \min \left\{ G_i(t); \hat{A}_i(t) \right\}$$

A Figura 6.1 ilustra a construção de  $U_i(t)$  para o caso do tráfego policiado pelo algoritmo do Balde Furado Fractal. É possível verificar que a curva  $G_i(t)$  intercepta  $\hat{A}_i(t)$  sempre que a rede for estável. De fato, Parekh e Gallager [33] provaram que  $G_i(t)$  intercepta  $\hat{A}_i(t)$  em  $t \leq \sum_{m \in P(i)} e_i^m$ , se as sessões forem reguladas pelo algoritmo do Balde Furado. A prova estabelecida por estes autores é válida também para o caso de sessões reguladas pelo algoritmo do Balde Furado Fractal e, por consequência, no mínimo uma solução para  $Q_i^*$  (e para  $D_i^*$ ) pode ser obtida:

**Lema 6.13.** *Para toda sessão  $i$ ,*

$$Q_i^* \leq \max_{t \geq 0} \left\{ \hat{A}_i(t) - U_i(t) \right\} \quad (6.8)$$

*e o valor de  $D_i^*$  pode ser obtido utilizando a aproximação dada por (5.6).*

Dado que  $U_i(t) = \min \left\{ G_i(t); \hat{A}_i(t) \right\}$ , a relação (6.8) pode ser reescrita como  $Q_i^* \leq \max_{t \geq 0} \left\{ \hat{A}_i(t) - G_i(t) \right\}$ . Esta desigualdade é válida para sessões reguladas pelo algoritmo do Balde Furado [33, Theorem 3]. A prova correspondente pode ser utilizada para demonstrar sua validade também para sessões reguladas pelo algoritmo Balde Furado Fractal.

É possível ainda assegurar a unicidade da solução de (6.8) :

**Teorema 6.14.** *Se o problema de otimização representado pela relação (6.8) tem solução, então esta solução é única.*

*Demonstração.* Seja  $t'$  o instante no qual  $U_i(t)$  intercepta  $\hat{A}_i(t)$ . Por construção, a função  $U_i(t)$  é estritamente côncava no intervalo  $[0; t']$ . Dado que  $\hat{A}_i(t)$  é estritamente convexa neste intervalo, a diferença  $\hat{A}_i(t) - U_i(t)$  também é estritamente convexa, o que garante a unicidade da solução de (6.8).  $\square$

#### Algoritmo para cálculo dos limitantes de $Q_i^*$ e $D_i^*$

O algoritmo para a obtenção de limitantes para o *backlog* e para o atraso fim-a-fim em uma rede de servidores GPS-CRST é apresentado no Algoritmo 3. Este algoritmo é capaz de obter limitantes para todas as sessões em uma rede GPS-CRST estável, o que constitui uma vantagem em relação a uma outra abordagem proposta por Zhang et al. [48], que assume que todas as sessões são localmente estáveis.

O algoritmo proposto é dividido em duas partes, e opera da seguinte maneira: primeiro, todas as inclinações  $s_j^{i,m}$  e durações  $d_j^{i,m}$  são calculadas para todas as sessões na rede. Para cada servidor  $m$ , este cálculo é realizado de modo similar ao caso de um servidor isolado.

Na segunda etapa, são calculados os limitantes para o *backlog* e o atraso fim-a-fim de cada uma das sessões na rede. Para cada sessão  $i$ , é definido o conjunto  $E_i$ , do qual os segmentos são tomados em ordem crescente de inclinação. Para cada segmento, o máximo *backlog* é identificado. Se ele é atingido dentro do segmento, então  $Q_i^*$  e  $D_i^*$  são calculados e o procedimento é repetido para outra sessão. O algoritmo termina quando  $Q_i^*$  e  $D_i^*$  forem determinados para todas as sessões na rede.

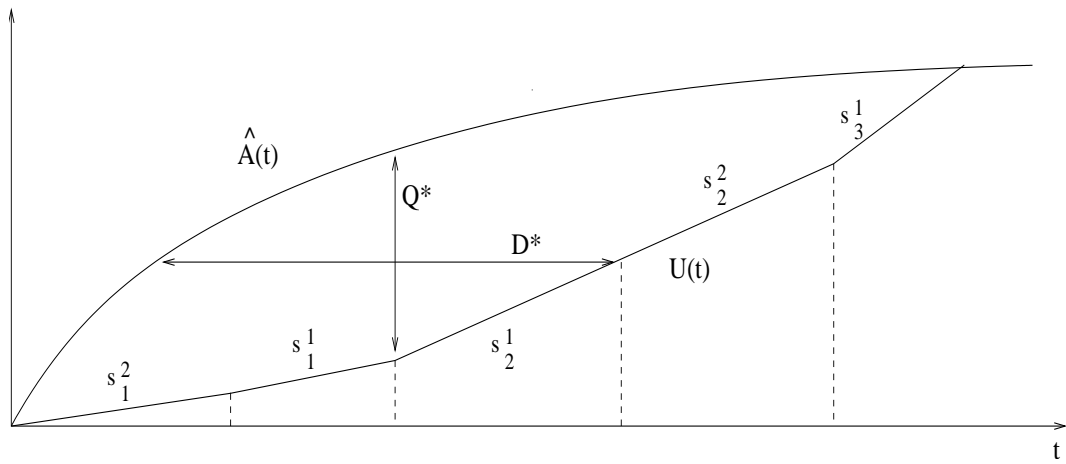
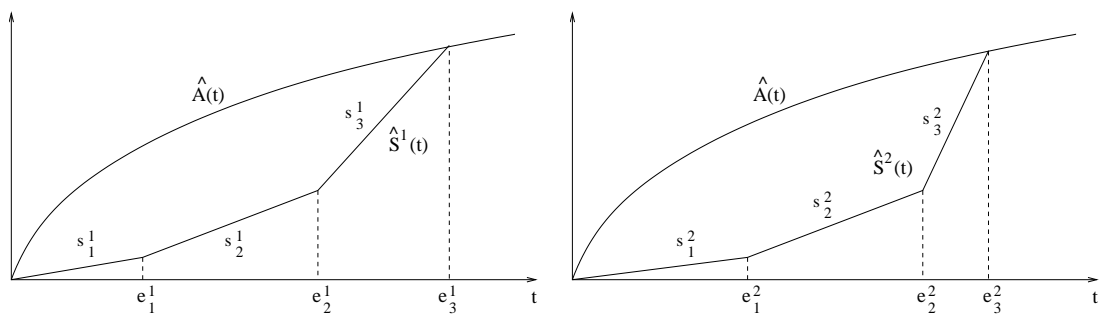


Figura 6.1: Exemplo de construção da curva  $U_i(t)$  considerando que  $P(i)$  é composto por dois servidores.

---

**Algoritmo 3** Algoritmo para o cálculo de  $Q_i^*$  e  $D_i^*$  utilizando a Curva de Serviço Universal.

---

```

for all servidor  $m$  do
   $e_0^m \leftarrow 0$ ;
  for  $\forall i \in I(m)$ ,  $\gamma_i^m \leftarrow 0$ ;
   $j \leftarrow 1$ ;
   $\mathbf{V} \leftarrow$ Conjunto das sessões em  $m$ ;
  {Calcule  $s_j^{i,m}$  para toda sessão  $i \in I(m)$ .}
  loop
    for  $\forall i \notin \mathbf{V}$ ,  $s_j^{i,m} \leftarrow \rho_i + H_i \psi_i e_{j-1}^{H_i-1}$ ;
    for  $\forall i \in \mathbf{V}$ ,  $s_j^{i,m} \leftarrow \frac{\phi_i^m}{\sum_{k \in \mathbf{V}} \phi_k^m} \left( r^m - \sum_{k \notin \mathbf{V}} s_k^{i,m} \right)$ ;
     $\mathbf{C} \leftarrow \left\{ i \in \mathbf{V} \mid s_j^{i,m} > \rho_i \right\}$ 
    if  $\mathbf{C} = \emptyset$  then
      break;
    end if
    {Determine qual sessão termina seu ciclo ativo primeiro.}
    for  $\forall i \in \mathbf{C}$  do
      Encontre  $t_i$  tal que  $\left( \rho_i - s_i^{i,m} \right) t_i + s_j^{i,m} e_{j-1}^m - \gamma_i^m + \psi_i t_i^{H_i} = 0$ ;
    end for
     $e_j^m \leftarrow \min_i t_i$ ;
    for  $\forall i \in \mathbf{C}$  do
       $d_j^{i,m} \leftarrow e_j^m - e_{j-1}^m$ ;
       $\gamma_i^m \leftarrow \gamma_i^m + s_j^{i,m} d_j^{i,m}$ ;
    end for
     $\mathbf{V} \leftarrow \mathbf{V} - \{ \arg \min_i t_i \}$ ;
     $j \leftarrow j + 1$ ;
  end loop
end for
{Determina limitantes para  $Q_i^*$  e  $D_i^*$  para toda }
{sessão  $i$  na rede.}
for all sessão  $i$  do
   $E_i = \bigcup_{m \in P(i)} \bigcup_j \left\{ \left( s_j^{i,m}; d_j^{i,m} \right) \right\}$ 
   $\gamma_i \leftarrow 0$ ;  $\tau \leftarrow 0$ ;
  loop
     $j \leftarrow$ Elemento cuja inclinação é a menor em  $E_i$ ;
     $(s_j; d_j) \leftarrow$ Inclinação e duração do elemento  $j$ ;
    Remove o elemento  $j$  de  $E_i$ ;
     $t^* \leftarrow \left( \frac{s_j - \rho_i}{\psi_i H_i} \right)^{\frac{1}{H_i-1}}$ 
     $t^* \leftarrow \max(\tau, t^*)$ ;
    if  $t^* < \tau + d_j$  then
       $Q_i^* \leftarrow (\rho_i - s_j) t^* - \gamma_i + s_j \tau + \psi_i t^{*H_i}$ ;
       $D_i^* \leftarrow Q_i^* \left( \min_{m \in P(i)} \frac{\phi_i^m}{\sum_{k=1}^N \phi_k^m} \right)^{-1}$ ;
      break; {Executa para a próxima sessão.}
    else
       $\gamma_i \leftarrow \gamma_i + s_j d_j$ ;
       $\tau \leftarrow \tau + d_j$ ;
    end if
  end loop
end for

```

---

### 6.3 Exemplo numérico

Um exemplo numérico da aplicação do Algoritmo 3 é apresentado nesta seção. Neste exemplo, o limitante para o atraso fim-a-fim obtido pelo algoritmo é comparado com o atraso obtido via simulação, e a validade do limitante é confirmada.

A Figura 6.2 apresenta uma rede *Tandem* composta por quatro nós servidores. Três sessões (Sessões 1, 2 e 3) entram na rede no Servidor 1, passam através dos Servidores 2 e 3, e deixam a rede após o Servidor 4. No Servidor 2, duas outras sessões (Sessões 4 e 5) entram na rede. A Sessão 4 deixa a rede após o Servidor 3, e a Sessão 5 a deixa após o Servidor 2. No Servidor 3, a Sessão 6 entra na rede, e a deixa após este mesmo servidor. Os parâmetros de tráfego e os fatores de ponderação GPS para todas as sessões são mostrados na Tabela 6.1.

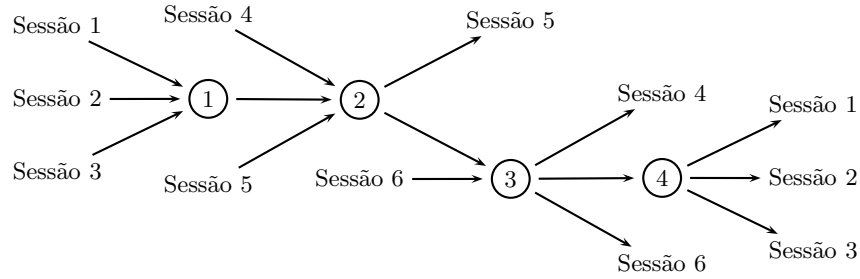


Figura 6.2: Rede utilizada no exemplo.

Sessão	$\rho_i$	$\sigma_i$	$k_i$	$H_i$	$\phi_i$
1	0,20	0,15	5,25	0,85	0,20
2	0,40	0,30	5,25	0,75	0,40
3	0,30	0,20	5,25	0,70	0,30
4	0,35	0,20	5,25	0,80	0,35
5	0,15	0,10	5,25	0,75	0,15
6	0,20	0,15	5,25	0,65	0,20

Tabela 6.1: Parâmetros utilizados no exemplo.

As Figuras 6.3 a 6.5 mostram o atraso fim-a-fim para as Sessões 1, 2 e 3 em função da utilização dos enlaces da rede. Os resultados de simulação foram obtidos utilizando o pacote Matlab/Simulink, no qual foi implementada a rede GPS alimentada por séries temporais artificiais com  $10^6$  amostras e exatamente conformes desde o instante zero. Pode-se verificar que os limitantes obtidos por meio da análise proposta são efetivamente válidos e próximos dos valores obtidos na simulação. A diferença entre os limitantes obtidos por meio do algoritmo e os valores obtidos na simulação se deve fato da curva de serviço universal  $U_i(t)$  não representar de modo exato o serviço fornecido a cada sessão, mas constituir apenas um limitante inferior a esta grandeza. Além disso, a precisão deste limitante depende da precisão das curvas  $\hat{S}_i(t)$ , obtidas em cada nó ao longo de  $P(i)$ . Este fato permite explicar o motivo da maior precisão obtida para o limitante da Sessão 3 em relação, por exemplo, ao limitante da Sessão 1: dados os parâmetros de tráfego e a parametrização GPS apresentadas na Tabela 6.1, a Sessão 3 sempre termina

seu ciclo ativo antes da Sessão 1. Desta forma, a construção da curva  $\hat{S}_i(t)$  em cada nó requer um número menor de segmentos no caso da Sessão 3 e, por isso, a aproximação da curva de serviço por  $\hat{S}_i(t)$  é mais precisa. Este fato se reflete na precisão da curva  $U_i(t)$  e, conseqüentemente, na precisão dos limitantes obtidos pelo algoritmo.

Para os mesmos parâmetros de tráfego, resultados similares foram obtidos para simulações utilizando seqüências fBm com  $10^6$  amostras, geradas utilizando o método de Mandelbrot [5] (Figuras 6.6 a 6.8). Pode-se notar que os limitantes obtidos por meio da análise da Curva de Serviço Universal continuam válidos, embora sejam menos precisos que no caso anterior, devido ao fato das seqüências não serem exatamente conformes.

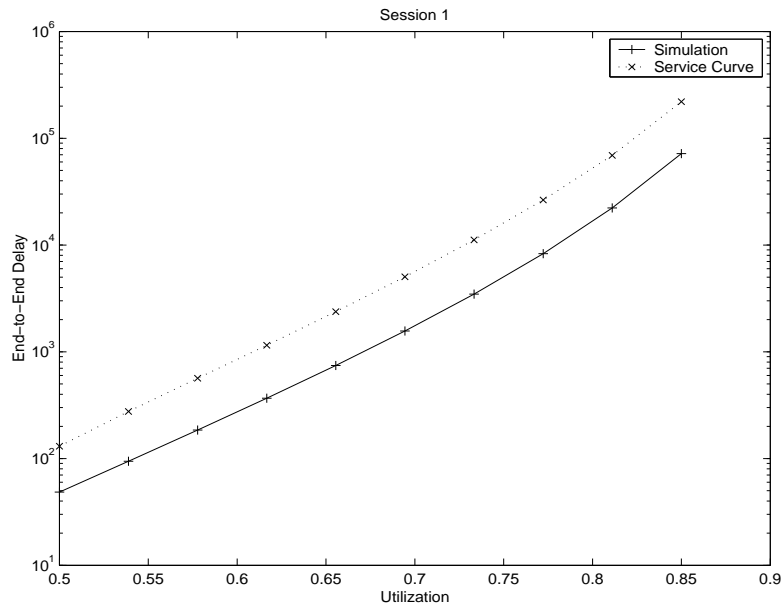


Figura 6.3: Comparação entre o limitante do atraso fim-a-fim da Sessão 1 e os valores obtidos por simulação, sob tráfego exatamente conforme.



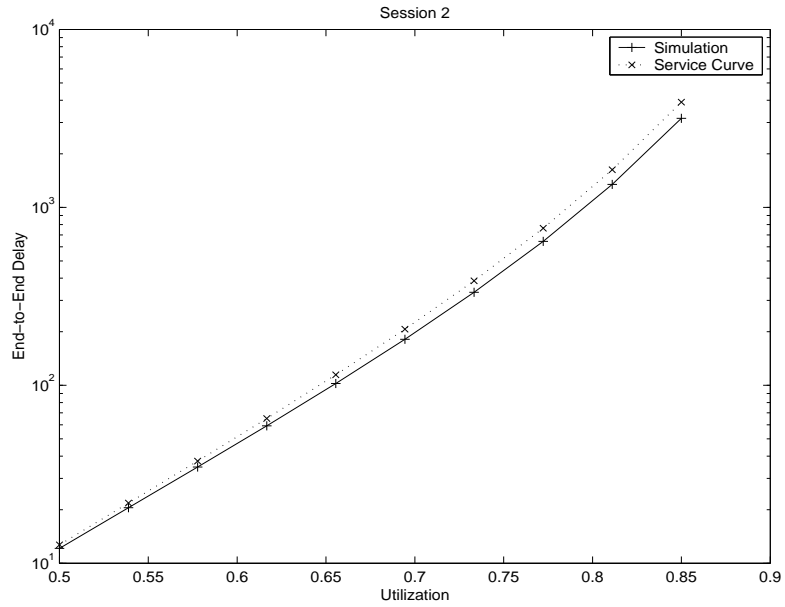


Figura 6.4: Comparação entre o limitante do atraso fim-a-fim da Sessão 2 e os valores obtidos por simulação, sob tráfego exatamente conforme.

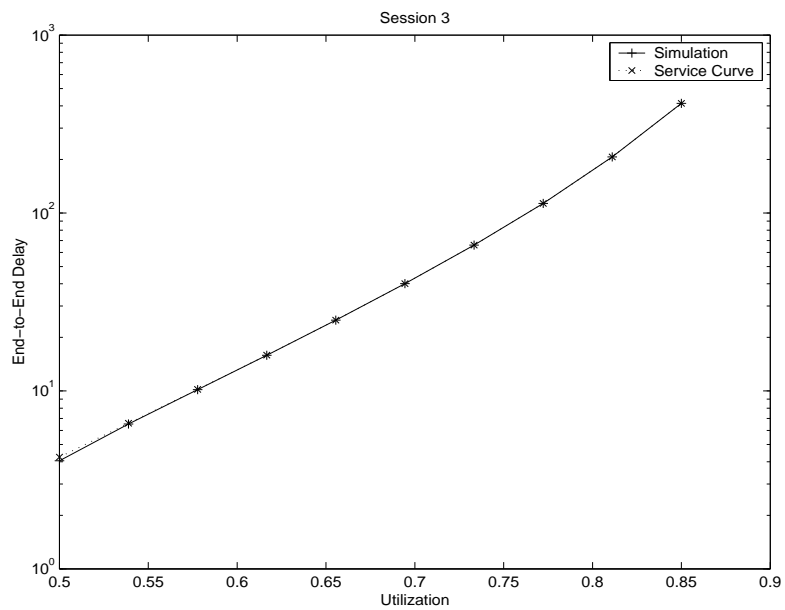


Figura 6.5: Comparação entre o limitante do atraso fim-a-fim da Sessão 3 e os valores obtidos por simulação, sob tráfego exatamente conforme.

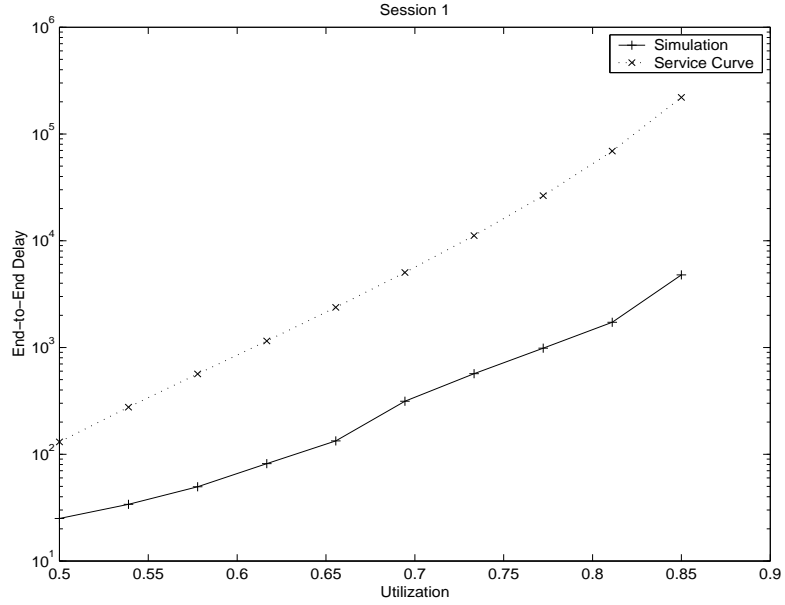


Figura 6.6: Comparação entre o limitante do atraso fim-a-fim da Sessão 1 e os valores obtidos por simulação, utilizando seqüências fBm.

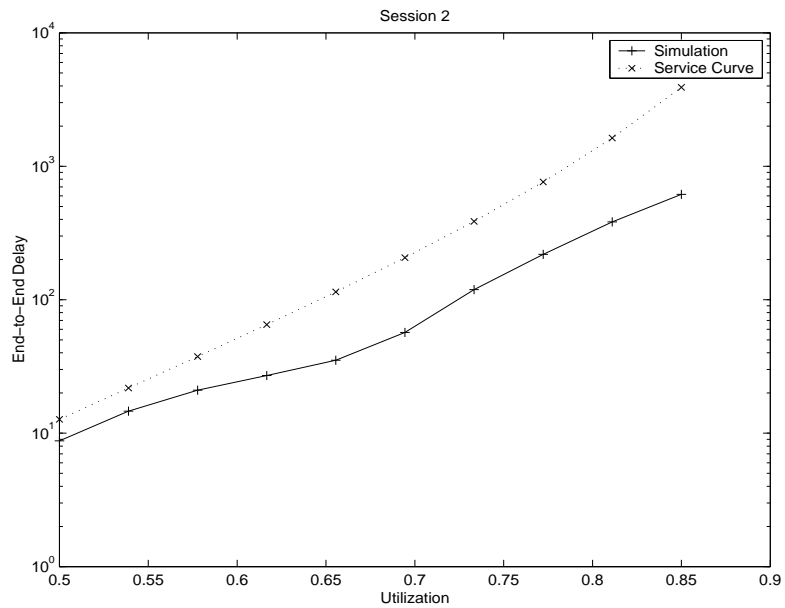


Figura 6.7: Comparação entre o limitante do atraso fim-a-fim da Sessão 2 e os valores obtidos por simulação, utilizando seqüências fBm.

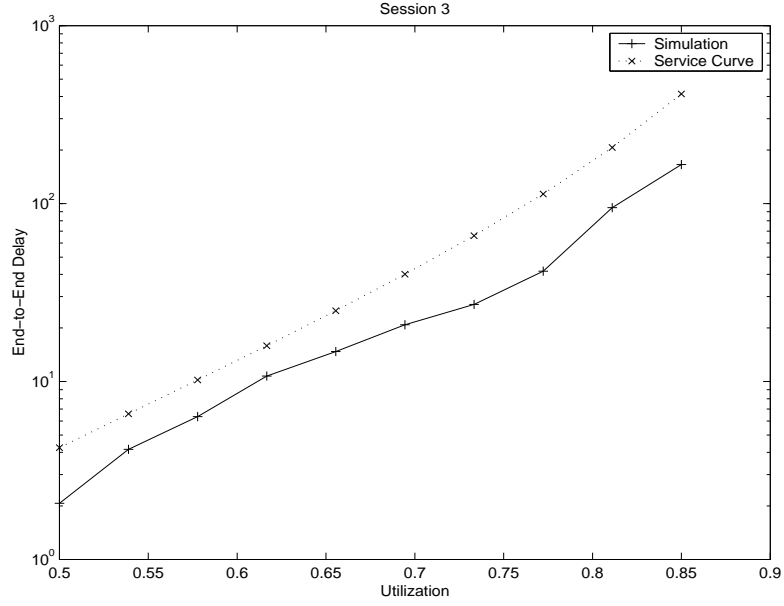


Figura 6.8: Comparação entre o limitante do atraso fim-a-fim da Sessão 3 e os valores obtidos por simulação, utilizando seqüências fBm.

## 6.4 Considerações finais

### 6.4.1 Atraso de propagação nos enlaces

Na análise desenvolvida nos capítulos anteriores, considerou-se que o atraso de propagação nos enlaces da rede é desprezível. Se, no entanto, o atraso de propagação de cada enlace é constante, Parekh e Gallager propõem uma maneira de incorporá-lo à análise [33]. Seja  $\delta_{m-1,m}$  o atraso de propagação introduzido pelo enlace que liga o nó  $m-1$  ao nó  $m$ , ambos pertencentes ao caminho de uma sessão  $i$ . O atraso introduzido pelo enlace pelo qual o tráfego desta sessão entra na rede é representado por  $\delta_{0,1}$ . Neste caso, o Lema 6.3 deixa de ser válido, e o tráfego dentro da rede passa a ser caracterizado por:

$$\begin{aligned}
 A_i^m(\tau; t) &= S_i^{m-1}(\tau - \delta_{m-1,m}; t - \delta_{m-1,m}) \\
 &= \hat{A}_i(\tau - (m-1)\delta_{m-1,m}; t - (m-1)\delta_{m-1,m})
 \end{aligned}$$

Desta forma, pode-se verificar que o limitante de atraso fim-a-fim que inclui o atraso de propagação dos enlaces é dado por:

$$D_{i(prop)}^* \leq D_i^* + \sum_{m \in P(i)} \delta_{m-1,m} \quad (6.9)$$

sendo  $D_i^*$  o limitante obtido quando os atrasos de propagação são desconsiderados. Por outro lado, pode-se considerar que, em um dado instante, o máximo volume de tráfego da sessão  $i$  que estará em transmissão no enlace é dado por:

$$q_{m-1,m} = r^{m-1} \delta_{m-1,m}$$

Sendo  $m_1$  o primeiro nó ao longo de  $P(i)$ , o máximo volume de tráfego da sessão  $i$  no interior da rede é, então, limitado por:

$$Q_{i(prop)}^* \leq Q_i^* + \sum_{m \in P(i) - \{m_1\}} r^{m-1} \delta_{m-1,m} \quad (6.10)$$

#### 6.4.2 Limitantes para variação de atraso

Efetivamente, as disciplinas de serviço com conservação de trabalho são inadequadas ao controle da variação de atraso fim-a-fim de uma sessão [20, 47]. No entanto, é possível estabelecer um limitante (conservador) para este índice de desempenho:

$$\Delta D_i^* \leq D_i^* \quad (6.11)$$

i.e., a variação de atraso de uma sessão  $i$  qualquer é limitada pelo máximo atraso daquela sessão.

## Capítulo 7

# Conclusões

Parekh e Gallager [33] demonstraram que a disciplina de serviço GPS é capaz de prover garantias de atraso quando o tráfego das sessões é policiado pelo algoritmo do Balde Furado. No entanto, tal algoritmo não é adequado ao policiamento de tráfego auto-similar, em especial na presença de dependências de longa duração [14, 26].

A principal contribuição do presente trabalho é, portanto, a análise do GPS sob tráfego auto-similar. A partir desta análise, foi desenvolvido um algoritmo capaz de calcular limitantes para o *backlog* e para o atraso introduzido pelo servidor GPS no tráfego de cada uma das sessões, quando estas são policiadas pelo algoritmo do Balde Furado Fractal. Os resultados obtidos para um servidor GPS isolado foram, então, aplicados na análise de uma rede de servidores GPS com topologia arbitrária. Parekh e Gallager mostraram que, sendo o tráfego das sessões policiado pelo algoritmo do Balde Furado, condições suficientes para garantir estabilidade poderiam ser estabelecidas caso a parametrização dos servidores pertencesse a uma classe denominada Tratamento de Sessões Relativamente Consistente (*Consistent Relative Session Treatment*, CRST). Resultados similares foram obtidos, neste trabalho, para o caso do tráfego policiado pelo algoritmo do Balde Furado Fractal. Além disso, foi apresentado um algoritmo para o cálculo de limitantes para o *backlog* e para o atraso fim-a-fim correspondente a cada uma das sessões em uma rede GPS-CRST estável. Ressalta-se a fundamental importância deste cálculo no contexto das redes de serviço garantido.

Diversas extensões podem ser apontadas para este trabalho. Por exemplo, na análise de redes GPS considerou-se que o caminho percorrido pelo tráfego de uma determinada sessão é único e previamente estabelecido. Dado que esta consideração impõe restrições do ponto de vista da Engenharia de Tráfego, uma possível extensão seria a análise de redes GPS nas quais o caminho percorrido pelo tráfego de uma sessão não é único.

Pode-se citar ainda, como extensão para este trabalho, a análise de desempenho das variantes pacote-a-pacote do GPS, tais como o PGPS [32]. Neste sentido, é possível mostrar que o comportamento destas disciplinas se aproxima daquele apresentado pelo próprio GPS conforme o tamanho dos pacotes torna-se desprezível diante da

capacidade dos servidores e dos enlaces de transmissão. No entanto, quando os pacotes têm tamanho significativo, devem ser levados em consideração pelo menos dois efeitos até agora desprezados: *(i)* os pacotes são servidos de modo não-preemptivo, e *(ii)* os pacotes são considerados disponíveis para transmissão somente após a chegada do último bit. As linhas gerais para a análise do PGPS, em particular quando o tráfego é policiado pelo algoritmo do Balde Furado, são apontadas por Parekh e Gallager em [33].

# Referências Bibliográficas

- [1] P. Abry e D. Veitch. Wavelet analysis of long-range dependent traffic. *IEEE Trans. Inform. Theory*, 44(1):2–15, Jan. 1998.
- [2] J. Beran, R. Sherman, M. S. Taqqu, e W. Willinger. Long range dependence in VBR video traffic. *IEEE Trans. Commun.*, 43:1566–1579, 1995.
- [3] S. Borst, O. Boxma, e P. Jelenković. Asymptotic behavior of Generalized Processor Sharing with long-tailed traffic sources. In *Proc. IEEE/ INFOCOM*, pp. 912–921, 2000.
- [4] S. Borst, O. Boxma, e P. Jelenković. Generalized Processor Sharing with long-tailed traffic sources. In P. Key e D. Smith, editores, *Telnetraffic engineering in a competitive world, Proc. ITC-16*, pp. 345–354, 2000.
- [5] M. Chi, E. Neal, e G. Young. Practical applications of fractional brownian motion and noise to synthetic hydrology. *Water Resources Research*, 9:1523–1533, Dec. 1973.
- [6] J.-F. Coeurjolly. Simulation and identification of the fractional Brownian motion: a bibliographical and comparative study. Technical report, LMC-IMAG, Grenoble, France, 2000.
- [7] R. Cohen, N. L. S. Fonseca, e M. Zukerman. Traffic management and control. In M. Talipamula e B. Kasnabesh, editores, *Multimedia Communications Networks*. Artech House, 1998.
- [8] M. E. Crovella. Self-similarity in WWW traffic: evidence and possible causes. *IEEE Trans. Networking*, 5(6):835–845, Dec. 1997.
- [9] R. Cruz. A calculus for network delay, part I : Network elements in isolation. *IEEE Trans. Inform. Theory*, 37(1):114–131, Jan. 1991.
- [10] R. Cruz. A calculus for network delay, part II : Network analysis. *IEEE Trans. Inform. Theory*, 37(1):132–141, 1991.
- [11] N. G. Duffield e N. O’Connell. Large deviation and overflow probabilities for the general single-server queue, with applications. *Proc. Cambridge Philosophical Society*, (118):363–374, 1995.

- [12] P. Dupuis e K. Ramanan. A Skorokhod problem formulation and large deviation analysis of a processor sharing model. Technical report, Division of Applied Mathematics, Brown University, 1997.
- [13] N. L. S. Fonseca, G. S. Mayor, e C. A. Viana Neto. Policing and multiplexing of self-similar sources. Relatório Técnico IC99-01, Instituto de Computação, Universidade Estadual de Campinas, 1999.
- [14] N. L. S. Fonseca, G. S. Mayor, e C. A. Viana Neto. On the equivalent bandwidth of self-similar sources. *ACM Trans. on Modeling and Computer Simulation*, 10(2):104–124, Apr. 2000.
- [15] M. W. Garret e W. Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *Proc. ACM/SIGCOMM*, pp. 269–280, 1994.
- [16] C. Goldie e C. Klüppelberg. Subexponential distributions. In *A practical guide to heavy tails: statistical techniques for analyzing heavy tailed distributions*, pp. 435–459, Boston, 1998. Birkhäuser.
- [17] D. Heymand e T. Lakshmon. What are the implications of long range dependence for VBR video traffic engineering ? *IEEE/ACM Trans. Networking*, 4:301–317, June 1996.
- [18] G. A. Hirchoren. *Predição e estimação de parâmetros de processos auto-similares para redes de faixa larga*. Tese de doutorado, Universidade Estadual de Campinas, 1999.
- [19] H. E. Hurst, R. Black, e Y. Simaika. *Long-term storage: an experimental study*. Constable, London, 1965.
- [20] S. Keshav. *An engineering approach to computer networking: ATM networks, the Internet and the telephone network*. Addison Wesley, 1997.
- [21] W. Lau, A. Erramilli, J. L. Wang, e W. Willinger. Self-similar traffic generation: the random midpoint displacement algorithm and its properties. In *Proc. IEEE/ICC*, pp. 54–69, 1995.
- [22] W. E. Leland, W. Willinger, M. S. Taqqu, e D. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Networking*, 2(1):1–15, Feb. 1994.
- [23] B. B. Mandelbrot. *The fractal geometry of nature*. W.H. Freeman and Company, New York, 1982.
- [24] B. B. Mandelbrot e J. W. van Ness. Fractional brownian motions, fractional noises and applications. *SIAM Review*, 10:422–437, Oct. 1968.
- [25] G. Mayor e J. A. Silvester. Providing QoS for long-range dependent traffic. In *The 7th IEEE Computer-Aided Modeling, Analysis and Design of Communications Link and Networks*, pp. 19–28, 1998.
- [26] G. S. Mayor e J. A. Silvester. Time scale analysis of an ATM queueing system with long-range dependent traffic. In *Proc. IEEE/INFOCOM*, pp. 205–212, 1997.



- [27] M. Montgomery e G. de Veciana. On the relevance of time scales in performance oriented traffic characterizations. In *Proc. IEEE/INFOCOM*, pp. 513–520, 1996.
- [28] P. R. Morin. *The impact of self-similarity on network performance analysis*. Ph.d. dissertation, Carleton University, Dec. 1995.
- [29] I. Norros. A storage model with self-similar input. *Queueing Syst.*, 16:387–396, 1994.
- [30] I. Norros. On the use of fractional brownian motion in the theory of connectionless networks. *IEEE J. Select. Areas Commun.*, 13(6):953–962, Aug. 1995.
- [31] I. Norros, P. Mannersalo, e J. Wang. Simulation of fractional brownian motion with conditionalized random midpoint displacement. *Adv. Performance Analysis*, 2(1):77–101, 1999.
- [32] A. K. Parekh e R. G. Gallager. A Generalized Processor Sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Trans. Networking*, 1(3):344–357, June 1993.
- [33] A. K. Parekh e R. G. Gallager. A Generalized Processor Sharing approach to flow control in integrated services networks: the multiple-node case. *IEEE/ACM Trans. Networking*, 2(2):137–150, Apr. 1994.
- [34] K. Park, G. Kim, e M. Crovella. On the relation between file sizes, transport protocols and self-similar network traffic. In *Proc. IEEE Int’l. Conf. Network Protocols*, pp. 171–180, Oct. 1996.
- [35] K. Park, G. Kim, e M. Crovella. On the effect of traffic self-similarity on network performance. In *Proc. SPIE Int’l. Conf. Perf. and Control of Network Sys.*, pp. 296–310, 1997.
- [36] K. Park e W. Willinger, editores. *Self-similar network traffic and performance evaluation*. Wiley, New York, 2000.
- [37] V. Paxson. Fast approximation of self-similar network traffic. Technical Report LBL-36750, Lawrence Berkeley Laboratories, 1995.
- [38] V. Paxson. Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic. *Computer Communication Review*, 27(5):5–18, 1997.
- [39] V. Paxson e S. Floyd. Wide area traffic: The failure of poisson modeling. *IEEE Trans. Networking*, 3(3):226–244, June 1995.
- [40] B. Ryu e A. Elwalid. The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities. In *Proc. ACM SIGCOMM*, 1996.

- [41] W. Stallings. *ISDN and Broadband ISDN with Frame Relay and ATM*. Prentice Hall, New Jersey, 3 edition, 1995.
- [42] M. van Uiter e S. Borst. Generalised Processor Sharing networks fed by heavy tailed traffic flows. In *Proc. IEEE/INFOCOM*, 2001.
- [43] C. A. Viana Neto. *Multiplexação e policiamento de tráfego auto-semelhante*. Dissertação de mestrado, Universidade Estadual de Campinas, 1999.
- [44] W. Willinger, M. S. Taqqu, R. Sherman, e D. V. Wilson. Self-similarity through high-variability: statistical analysis of ethernet LAN traffic at the source level. *IEEE/ACM Trans. Networking*, 5(1):71–86, Feb. 1997.
- [45] O. Yaron e M. Sidi. Performance and stability of communication networks via robust exponential bounds. *IEEE/ACM Trans. Networking*, 1:372–385, 1993.
- [46] O. Yaron e M. Sidi. Generalized Processor Sharing networks with exponentially bounded burstiness arrivals. In *Proc. IEEE/ INFOCOM*, pp. 628–634, 1994.
- [47] H. Zhang. Service disciplines for guaranteed performance service in packet switching networks. *Proc. IEEE*, 83(10):1374–1396, Oct. 1995.
- [48] Z.-L. Zhang, D. Towsley, e J. Kurose. Statistical analysis of the Generalized Processor Sharing scheduling discipline. *IEEE J. Select. Areas Commun.*, 13(6):1071–1080, Aug. 1995.